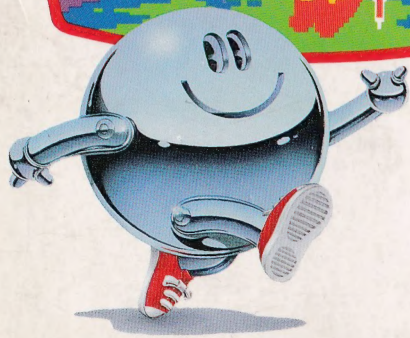
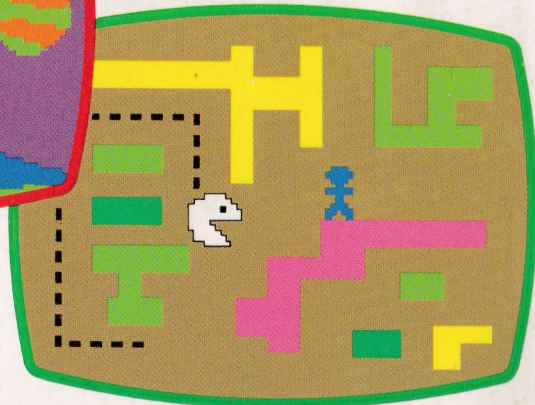
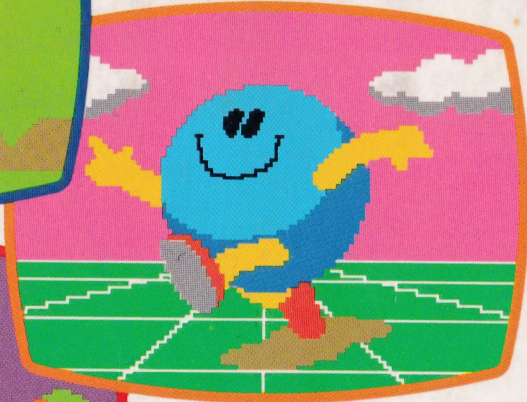
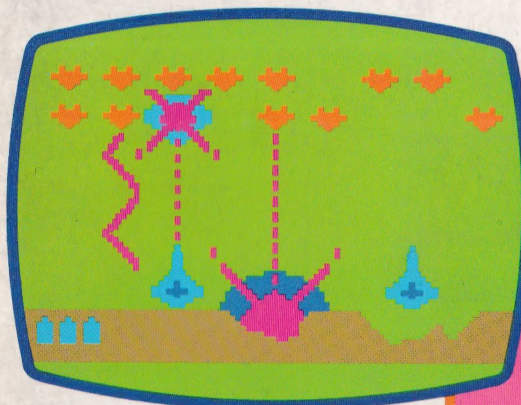
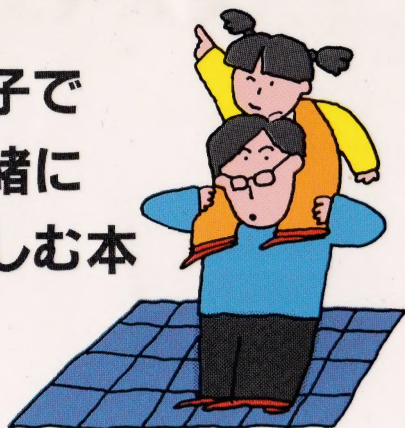
SEGA SK-1100による
ゲームとアニメーションのつくり方
(株)システムハウス・オプト／著
▶BASICでもこんなに楽しめる!
▶絵を描く、音を出す、ゲームができるプログラミングのコツ
日本実業出版社

山を描く
色をぬる
UFOをつくる
ビーム光線をつくる
カーレースの道路もOK！
テニスのラケットを動かす
……

＊BASICを使ったゲームとアニメのつくり方、プログラミングのコツがいっぱい。

＊SEGA BASICはMSX BASICとほとんど同じですから、MSXマシンを持っている人にも役立ちます。

＊実際にコンピュータをさわりながら読んでいってください。ボクだけのゲーム、ワタシだけのアニメがきっとできるようになります。

SEGA® SK-1100による

ゲームとアニメーションのつくり方

(株)システムハウス・オプト／著

日本実業出版社

♥がつくか、×がつくか？
相性診断なんていかが？

あみだくじ？ 迷路？ いえいえ、これはキミの運命というか、相性というか、とにかく運まかせの遊び、ゲームなんです。ひとつ、このプログラムを入力してプレイしてみてください。

```
10 REM *** アイショウ シンダ゛ン ***
20 CLS
30 V=15567:T=1:A=129:Q=252:C=0:D=0
40 CURSOR9,0:PRINT"      アイショウ シンダ゛ン
"
50 FOR Y=1 TO 18 :A$="|"
60 FOR X=6 TO 30 STEP8
70 IF Y=1 THEN GOSUB 370
80 CURSORX,Y:PRINTA$
90 IF Y=1 THEN CURSORX,19:PRINTB$
100 NEXT X,Y
110 FOR Y=5 TO 15 STEP 5
120 GOSUB 370
130 CURSOR 2,Y:PRINTA$+"───────┼──────────────┼───
─────────┼───────────┼──────"+B$
140 NEXT Y
150 CURSOR8,21:PRINT"                   50%   1
00%                       アイショウト゛"
160 VPOKE V,Q
170 FOR W=0 TO 10:NEXT W
180 VP=VPEEK(V+T)
190 VPOKE V,A
200 IF VP=146 THEN 260
210 IF VP=88 THEN T=T*-1:C=C+1
220 IF VP=246 THEN T=T*-1:C=C+1:D=D+1:
CURSOR14+D,22:PRINT"♥"
230 IF C=10 THEN 400
240 V=V+T
250 GOTO 160
260 Z=4:IF T=1ORT=-1 THEN Z=1
270 R=INT(RND(1)*3+Z)
280 ON R GOSUB 310,320,330,340,350,360
```

```
290 Q=251:A=128:IF T=1ORT=-1 THEN Q=25
2:A=129
300 GOTO 160
310 V=V+2:RETURN
320 V=V+40+T:T=40:RETURN
330 V=V-40+T:T=-40:RETURN
340 V=V+80:RETURN
350 V=V+T+1:T=1:RETURN
360 V=V+T-1:T=-1:RETURN
370 B=INT(RND(1)*2):A$="♥":B$="X"
380 IF B=1 THEN A$="X":B$="♥"
390 RETURN
400 CURSOR13,10:PRINT"PUSH ANY KEY"
410 FOR I=0 TO 2000
420 IF INKEY$="" THEN NEXTI
430 GOTO 20
```

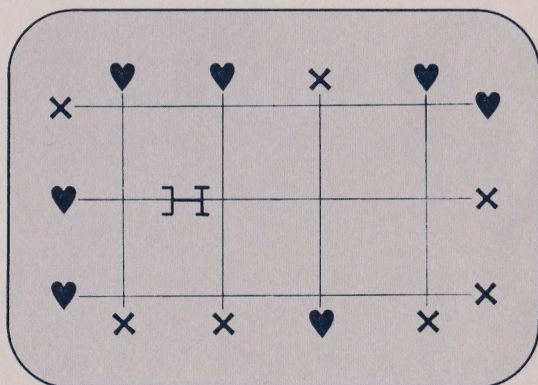
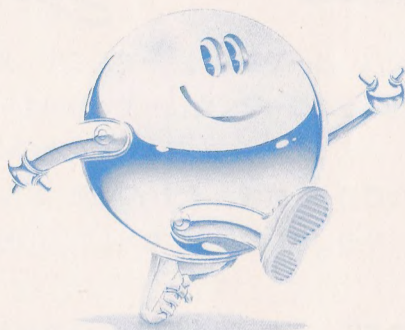
画面ですョ

SEGA® SK-1100による

ゲームとアニメーションのつくり方

(株)システムハウス・オプト／著

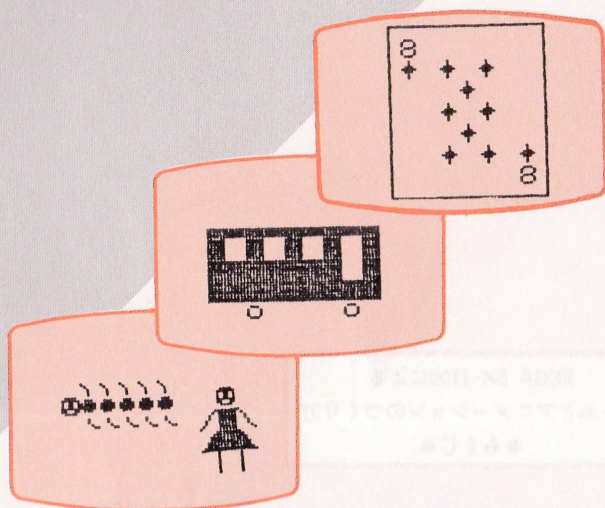
日本実業出版社

はじめに

グランプリレース、テニスゲーム、撃墜ゲーム、ハンターゲーム、ベースボールゲーム……こんな、ゲームセンターにあるようなゲームが、もし自分でつくれたらいいナァ……

UFO、パックマン、クルマ、ヒト、動物……こういうアニメもおもしろいけど、ワタシだけのアニメがつくれないかしら？……

こう思っている人はたくさんいると思います。ですが、〝どうやってつくったらいいかわからない〟と尻込みしているのではありませんか？

この本は、そういう人のために、ＢＡＳＩＣを使った、ゲームとアニメーションのつくり方を紹介したものです。〝えっ？　ＢＡＳＩＣでできるの!?〟という、ちょっと専門家もいるでしょう。ゲームセンターにあるゲームはマシン語という言葉を使っていますが、ＢＡＳＩＣだって立派にゲームができるんですよ。

また、この本ではＳＥＧＡ　ＳＫ-1100というコンピュータを例にしてＳＥＧＡ-ＢＡＳＩＣで説明していますが、これはＭＳＸ-ＢＡＳＩＣとほとんど同じですから、ＭＳＸコンピュータでも利用できます。

0　SEGA SK-1100はこんなマシン

1　簡単な絵を描く

UFOやパックマンがつくれる！　さあスタート!!

2　キャラクタが動く

タテ、ヨコ、ナナメ自由自在。ゲームらしくなってきたゾ！

3　キーでキャラクタを操作する

止めたり、動かしたり…キミだけの暗号をつくっちゃおう！

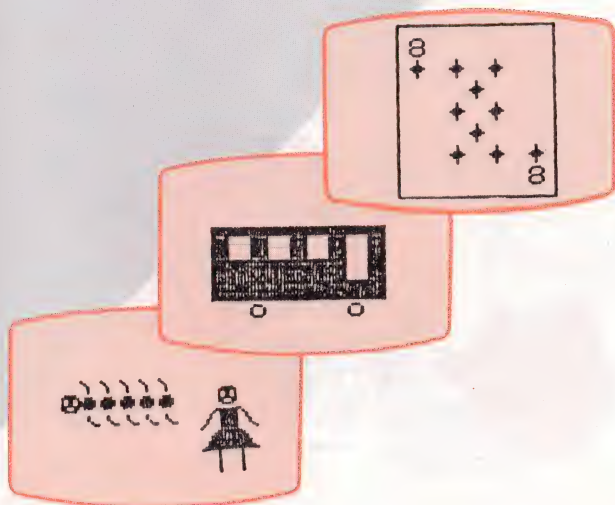
4　C　G（コンピュータ・グラフィック）にアタック！

ゲームのキャラクタ、背景づくりのテクニックをつかもう!!

5　もっともっとアニメらしく

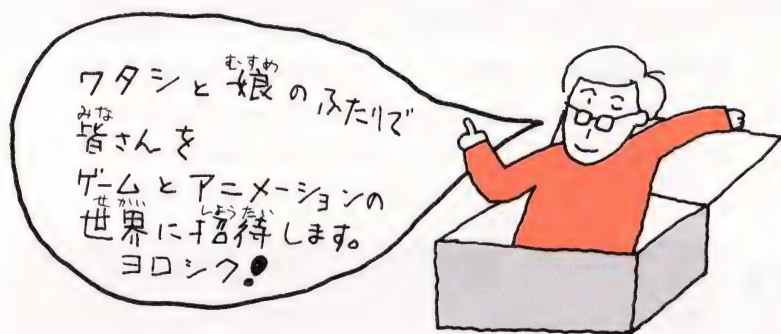
自分のつくったオリジナルアニメが画面の中を動きまわるゾ！

6　アイデアとテクニックの巻

ゲームをおもしろくするもしないも、キミのアイデア次第！

7 音を出す、音を使う

本格派ゲームの総仕上げだ！ むろん作曲もOK!!

8 基本プログラミングのコツ
ゲームづくりの総まとめ！　もちろんアニメもネ!!

9 スプライト機能でパワーアップ！
本格派ゲームの切り札！　おもしろさ倍増だよ!!

カバー装幀　小倉敏夫
カバーイラスト　尾崎英明
本文イラスト　江村留美子

0

SEGA SK-1100は こんなマシン

何事もはじめがカンジンといいます。もう、キミの頭の中ではUFOが飛びかうゲームや、色あざやかなアニメーションでいっぱいになっているかもしれませんね。

でもでも——。まずは、目の前にあるSK－1100に注目してください。ゲームづくりやアニメーションづくりをはじめる前に、SK－1100がどんなマシンなのかを紹介することにしましょう。

●各部の名称

SK-1100というコンピュータは、いくつかのものからできています。

★本　体

これは、BASICカートリッジを差しこむ本体のことです。ここが、コンピュータの頭脳にあたる部分です。

★キーボード

これは、文字や記号を打ちこむための装置(そうち)です。キーボードを見ると、たくさんのキーが並んでいますね。プログラムづくりに欠かせない部分といえます。

★テレビ

SK-1100は家庭用テレビでも使えます。

★プロッタプリンタ

もし、プログラムを紙に記録したい人は、プロッタプリンタを使うとよいでしょう。

★カセットレコーダー

もし、プログラムをカセットテープにしまっておきたい人は、カセットレコーダーを用意しましょう。

だいたい、以上で準備がそろいました。それぞれの接続方法は、取扱説明書を読んでください。

●電源ON

では、いよいよコンピュータを動かすことにしますが、電源を入れる前に、本体とキーボードの接続、テレビの接続、その他、プロッタプリンタやカセットテープレコーダーの接続がまちがっていないか、しっかりとつながっているかを確かめましょう。

接続がOKなら、BASICカートリッジを本体に差しこんでください。いよいよスタートです。

この場合、

① テレビの電源スイッチ

② 本体の電源のスイッチ

という順に、ONにします。

```
SEGA SC-3000 BASIC LEVEL 2 ver 1.0
Copyright 1984 (C) by MITEC

2043 Bytes free
Ready
```

すると、画面に上のような表示が出てきます。そうしたら、BASICを使いはじめてください。

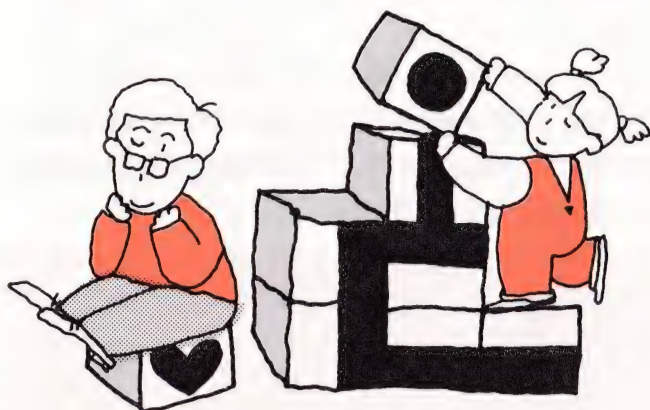
もし、この画面にならないときは、もう一度それぞれの装置の接続をたしかめ、また電源コードがコンセントに差しこんであるかどうかも、たしかめてください。

それでは、いよいよゲームとアニメの世界にすすんでいきましょう。

レッツ　ゴー！

1

簡単な絵を描く

UFOやパックマンがつくれる！さあスタート‼

スイッチオン！　する前に

キー上に並んでいる文字や図形を使って、簡単な絵を描いてみます。この絵は、ゲームの中に登場するUFOやパックマンなどのキャラクタとして、使うことができます。最初から、ふくざつなキャラクタをつくるのはむずかしいので、まず、簡単なものから始めていきましょう。

1 キャラクタを利用すれば簡単だ

キーボードを見ると、たくさんのキーが並んでいます。キーの表面には文字や図形が書かれていますが、これは**キャラクタ**と呼ばれています。

まず、キーの上の黄色で書かれている図形に注目してください。これは**グラフィックキャラクタ**とも呼び、文字や数字とはちょっと違った使い方をします。よく見ると、小さな円盤やクルマもあり、「年」「月」「日」という漢字まであります。

ゲームやアニメを始めるための第1歩は、画面上に絵を描くことですが、このキャラクタをそのまま利用するのが一番カンタンです。ものはためし、次のように打ってください。そして、CR(キャリッジリターン) キーをポン！　すると……。

＊1・1　まず入力してみよう

```
10 REM *** 1.1 ***
20 REM *** ｴﾝﾊﾞﾝ ﾀﾞﾖ ***
30 PRINT "⊕"
```

```
⊕
```

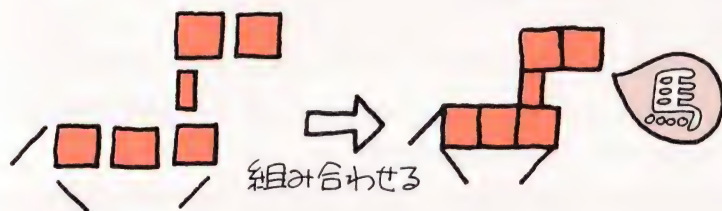
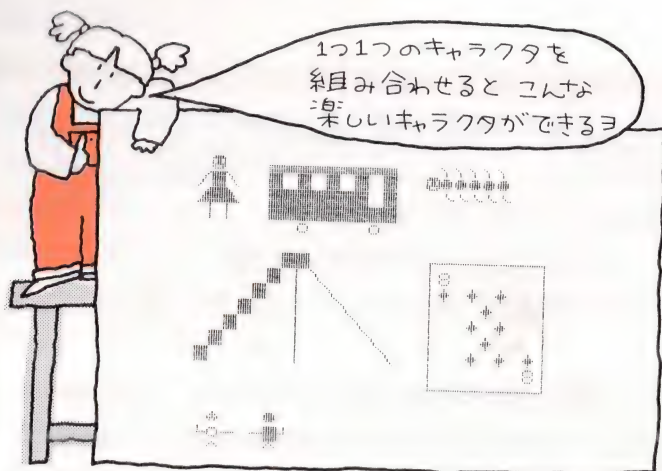
ちゃんと
円盤が描けた！

このほかにも、キーボードを見ると、ゲームやアニメに使えそうなキャラクタがありますね。ボールや弾（たま）として○と●、カーレース用の自動車として使えるのは工とH、ちょっと小さいけど人間として웃、あとはトランプゲームに使えそうな♠、♥、♦、♣。

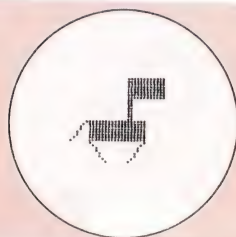
PRINT（プリント）　"○" とすると、○を表示し、　PRINT　"工" とやれば、画面にはちゃんと工を表示します。

でも "ボクは馬が登場するゲームをつくりたい" "私は女のコをつくってみたい" という人もいるでしょう。でも、馬や女のコはキーボードにはありません。こんなときは？

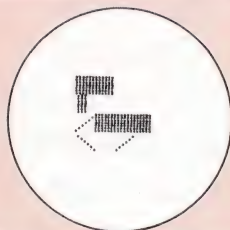
これも大丈夫。キーボードを見てください。◳、◲、⊞、◫といったキャラクタもありますし、■や▌、⧅、⧄といったものもあります。そうです、こういったキャラクタを組み合わせてつくってしまうのです。

簡単な馬ならば、■、▌、⧅、⧄の４つを組み合わせるだけで、いともカンタンにつくることができます。下の図は、馬の絵を各キャラクタに分解したものです。

実物とはちょっとかけ離れていますが、人間が웃なら、これは立派な馬ですね。この調子でキミの想像力（そうぞうりょく）をはたらかせて、いろんなモノをつくってみましょう。人間ひとつとってみても、女のコ、男のコ、歩いているところ、寝ているところ……何種類もつくることができます。

キャラクタを組み合わせて、ひとつのモノとして表示するには、PRINT文を使います。前ページの馬をつくるには、次のようなプログラムにします。とくに、スペースのとり方に注意してください。

＊1・2　馬をつくってみよう

```
5 REM *** 1.2 ***
8 REM *** ウマ ダヨ ***
10 PRINT "    ████"
20 PRINT "    █ "
30 PRINT "／█████"
40 PRINT " ＼ ／"
```

スペースをとらないと、下のようなちぐはぐな馬ができてしまいますよ。

＊1・3　マチガイやすいので注意

```
5 REM *** 1.3 ***
8 REM *** ヘンナ ウマ! ***
10 PRINT "████"
20 PRINT "█ "
30 PRINT "／█████"
40 PRINT "＼ ／"
```

2 キャラクタコードを使うというテもある

ところで……

PRINT "♠" とすると、画面には ♠ を表示します。

では、PRINT　CHR$(250)と打ってみてください。CHR$は「キャラクタ・ダラー」と読みますが、こうすると、やはり ♠ を表示するのです。

なぜでしょうか？　実はこの2つの命令は、まったく同じ働きをするのです。ということは、"♠"とCHR$(250)は、同じものといえますね。

このことは、"♠"だけではなく、"☻"とCHR$　(249)や"🯅"とCHR$(253)も同じです。

勘の鋭い人ならば、もうおわかりでしょう。(　)の中の数字に意味がありそうなことが。

そうです、キャラクタには、1つずつ決まった番号がついているのです。これを**キャラクタコード**と呼びます。たとえば、♠のキャラクタコードは250、☻は249となっています。

そして、CHR$(　　)のカッコ内にキャラクタコードを入れると、そのキャラクタを使うことができるのです。

もちろん、♠や☻のような「絵」だけにキャラクタコードがあるわけではありません。"A"とか"1"とかの文字や数字もキャラクタの一部ですから、これらにもキャラクタコードがちゃんとあります。

それぞれのキャラクタコードは、SK-1100専用の説明書のキャラクタコード表にのっています。次ページにその一部をのせました。説明書にはもちろん全部のっていますので、必要なときに見てみましょう。

163	」	179	ウ	195	テ	211	モ	227	◨	243	金
164	、	180	エ	196	ト	212	ヤ	228	▕	244	土
165	・	181	オ	197	ナ	213	ユ	229	■	245	♠
166	ヲ	182	カ	198	ニ	214	ヨ	230	◫	246	♥
167	ァ	183	キ	199	ヌ	215	ラ	231	⊟	247	♦
168	ィ	184	ク	200	ネ	216	リ	232	▒	248	♣
169	ゥ	185	ケ	201	ノ	217	ル	233	▒	249	☻
170	ェ	186	コ	202	ハ	218	レ	234	▚	250	[graphic]
171	ォ	187	サ	203	ヒ	219	ロ	235	◎	251	[graphic]
172	ャ	188	シ	204	フ	220	ワ	236	◉	252	[graphic]
173	ュ	189	ス	205	ヘ	221	ン	237	年	253	[graphic]
174	ョ	190	セ	206	ホ	222	゛	238	月	254	÷
175	ッ	191	ソ	207	マ	223	゜	239	日	255	

キャラクタコードは、必ずCHR$と組み合わせて使います。そして、頭にPRINTをつければ、そのキャラクタを画面に表示することができます。次のプログラムは、キャラクタコードを入力して、そのキャラクタを画面に表示するプログラムです。REM、CLS、INPUT、CODE、A$、GOTOといった言葉が出てきますが、今は理屈はヌキ。とにかくこのプログラムを入力してみてください。

＊1・4　いろいろなキャラクタを出してみよう

```
100 REM *** 1.4 ***
110 REM *** ﾅﾆ ｦ ﾀﾞｿｳｶﾅ ***
120 CLS
130 INPUT "CODE=?";C
140 A$=CHR$(C)
150 PRINT"CHR$(";C;")=";A$
160 GOTO 130
```

ここでRUNと打って CR キーを押すと、画面に“CODE=?”と表示してキャラクタコードをきいてきます。そうしたら、32から255までの数字を入力してみましょう。対応するキャラクタを表示します。次から次へと“CODE=?”ときいてきますので、そのつど違う数字を入れてみましょう。止めるには、BREAK キーを押します。

コンピュータと
お話ししながら……

```
CODE=?65
CHR$( 65)=A
CODE=?66
CHR$( 66)=B
CODE=?67
CHR$( 67)=C
CODE=?
Break in 130
```

3 自分のつくりたいキャラクタにアタック！

キー上のキャラクタをそのまま利用するのも楽しいものですが、もっと別のキャラクタがあったらナァと思う人もいるでしょう。

何しろ、SEGA SK-1100にあるキャラクタコードには数に限りがあります。これでは、自分のつくりたいキャラクタがない、と悲観してしまう人もいるはずです。

ところが、です。

ちゃんと自分のつくりたいキャラクタをつくることができるのです。

どうしてでしょう？

その秘密は、PATTERN文にあります。ためしに、次のプログラムどおり入力してみてください。

＊1・5　PATTERN文を使ってみよう

```
5 REM *** 1.5 ***
8 REM *** ナンダ゛ロウ? ***
10 PATTERN C#254,"FC84483030484884FC"
```

ん？ 何じゃコレは？ アルファベットと数字を並べただけじゃないか！と思う人もいるでしょう。マァマァ、そうあわてずに。このあと、RUN、 CR として ÷ を押してください。すると――。

```
⧗
```

へえーっ？

どうですか？　“⧖”と表示されるはずです。

これは、PATTERN文を使って“⧖”というキャラクタをつくったということなのです。

では、プログラムのタネあかしをしますね。

まず“C＃254”です。このうち254というのは、÷のキャラクタコードです。じゃ、÷を押すと、“⧖”を表示するというのは、ここで÷のキャラクタコードを指定しているからなんですね。

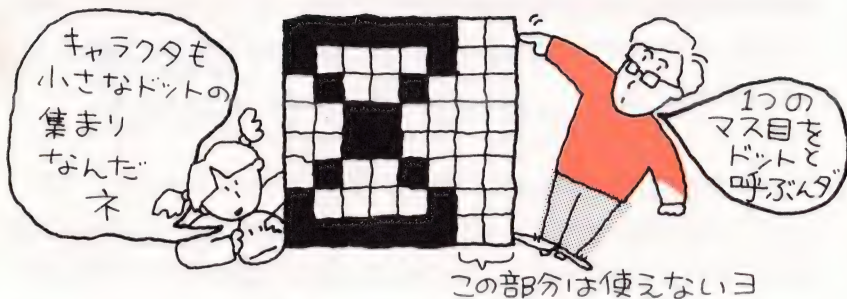
でも、絶対に÷だけかというと、そうではありません。あなたの好きなものを使っていいのですが、このさい、ふだんあまり使わないキャラクタのコードを指定すると便利だと思います。

もし“数学大好き人間”だったら、÷は計算するときよく使うでしょうから、こんな人だったら別のものを使うといいですね。

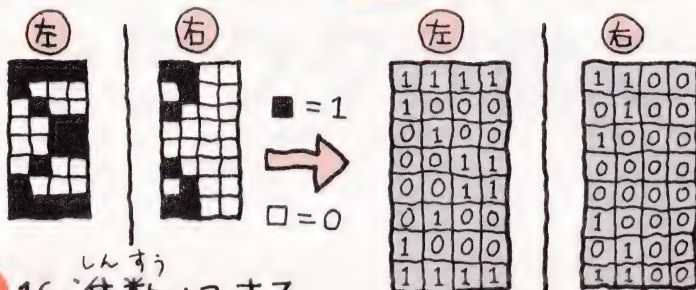
ただし、どんなキャラクタを使うにしても、キャラクタコードの前には“C＃”を、後ろには“，”を必ずつけてください。

次に、“FC844830304884FC”ですが、実はこれが“⧖”をつくっているのです。といっても、初めてコンピュータにさわる人だとわかりにくいと思いますので、実際に“⧖”のキャラクタをつくりながら説明していきます。

まず方眼紙を用意します。それに8×8のマス目を描いてください。このマス目1つ分を1ドットと呼びますが、ドットを1つずつぬりつぶして、あなたのつくりたいキャラクタを描いてみてください。ちょっとチグハグなキャラクタになることはがまんして、とにかくぬりつぶします。

どうですか？

キャラクタを描き終わったら、左右4ドットずつに分けてみましょう。実はこれがミソなのです。そして、■の部分は1、□の部分は0にして2進数の数字にします。たとえば、1列目は次のようになります。

2進数を、さらに16進数に変えます。1列目の左を16進数にすると"F"、右は"C"となりますね。左右を合わせると"FC"。"FC"が1列目のデータです。同じようにして、2列目以下のデータを出してみましょう。

"FC844830304884FC" というのは、1列目から8列目までのデータを左、右、左、右……と順番に並べたものです。

2進数	16進数	2進数	16進数	2進数	16進数
0000	0	0110	6	1100	C
0001	1	0111	7	1101	D
0010	2	1000	8	1110	E
0011	3	1001	9	1111	F
0100	4	1010	A		
0101	5	1011	B		

さあ、いよいよあなたがつくったキャラクタを、画面に表示してみましょう。プログラムを実行したら、“PRINT CHR$(　　)”を使って表示することができます。かっこの中には、あなたがつくったキャラクタを割り当てたキャラクタコードを入れてください。

表示できましたか？

このようなキャラクタのつくり方を利用したプログラムが、次のプログラムです。入力してみましょう。

＊1・6　ご存知、キン肉マン！

```
1005 REM *** 1.6 ***
1008 REM *** スーパ゚ーヒーロー キンニクマン ダ゛! ***
1010 PATTERNC#237,"00000C1C3C7C7460"
1020 PATTERNC#238,"8080E0F078FCDC8C"
1030 PATTERNC#239,"7C7C7C74381C0C0C"
1040 PATTERNC#240,"FC7CFCDC38F0E0E0"
1050 REM A$=237,B$=238,C$=239,D$=240
1060 A$=CHR$(237):B$=CHR$(238):C$=CHR$
(239):D$=CHR$(240)
1070 AA$=A$+B$+CHR$(31)+CHR$(29)+CHR$(
29)+C$+D$
1080 CURSOR18,5:PRINTAA$
1090 END
```

バイオマンや
ウルトラマンも
つくれるゾ！

2

キャラクタが動く

タテ、ヨコ、ナナメ自由自在。ゲームらしくなってきたゾ！

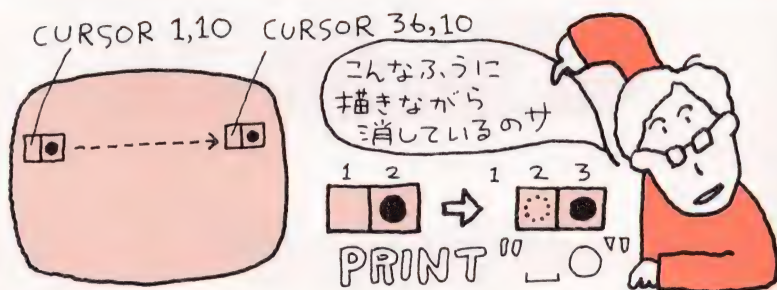
スイッチオン！　する前に

テレビゲームのＵＦＯやパックマン、それらはみんな、キャラクタを利用してつくられています。それでは、あちこちに動きまわるのは、なぜでしょうか。よく見ると、タテ、ヨコ、ナナメ自由自在に、微妙に違う動きをしています。ここでは、いろいろな動き方をプログラムにしてみました。

1 左から右へ動く

キャラクタがあちこちに動き回るというのは、どういうことなのでしょうか。さっそく、次のプログラムを入力して、実行してみましょう。

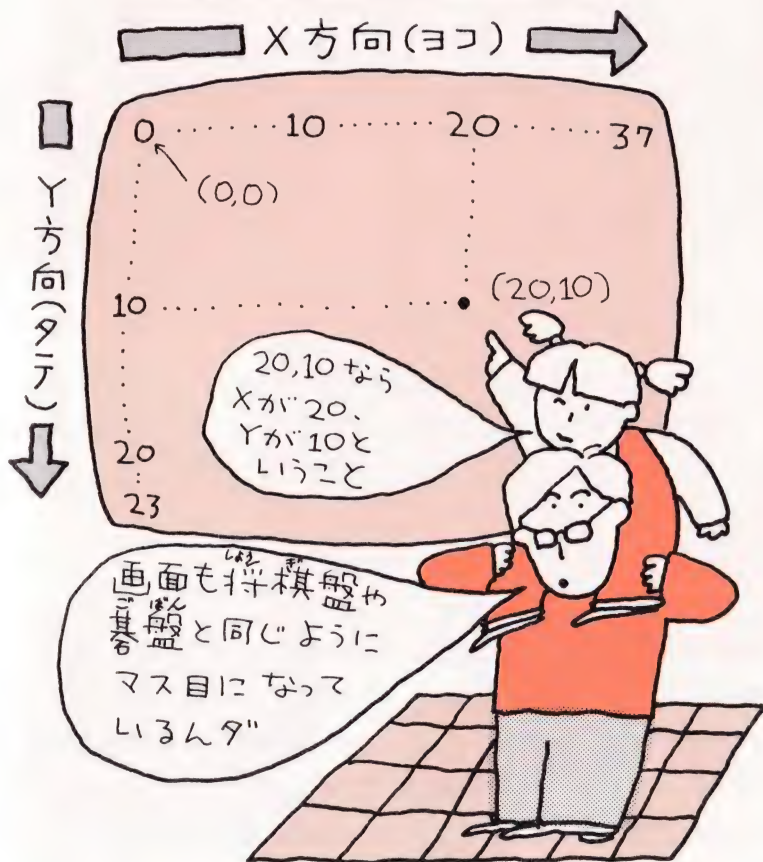
＊2・1　キャラクタを動かしてみよう

```
100 REM *** 2.1 ***
110 REM *** ﾐｷﾞ ﾍ ｺﾛｺﾛｺﾛ ***
120 CLS
130 FOR I=1 TO 36
140 CURSOR I,10
150 PRINT" ●";
160 NEXT I
170 GOTO 120
```

"●"というキャラクタが、画面を左から右へよこぎりますね。このプログラムは、いつまでも同じ動きをくり返します。途中でなるほどと思ったところで、BREAK を押してプログラムを止めましょう。

いったいぜんたい、どうしたらこんなふうにできるのかを、ちょっと説明しておきます。

キャラクタを表示しているのは、PRINTの働きです。そして、動くのは、CURSORの働きです。つまり、CURSORを使って、表示する位置を次から次へと変えていくことで、動いているように見せているのです。

また、行番号**150**でPRINT "　●" として●の左隣りに１字分スペースを入れているのは、直前に表示した "●" を消すためです。さらに、；（セミコロン）を忘れずにつけてください。

CURSOR 1，10の1，10って何？

2 右から左へ動く

ここで、前のページのプログラムの行番号130と150を、ちょっと変えてみましょう。

＊2・2　右から左へもOK！

```
100 REM *** 2.2 ***
110 REM *** ヒダリ ヘ スウウウ ***
120 CLS
130 FOR I=37 TO Ø STEP -1
140 CURSOR I,1Ø
150 PRINT"웃 ";
160 NEXT I
170 GOTO 120
```

右から左へよこぎる

웃 ←――――― 웃 ←――――― 웃

ヒトが
歩いているみたいだ！

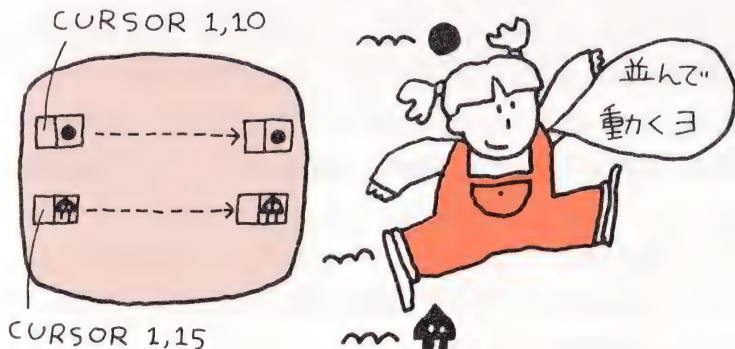
今度は"웃"が、右から左によこぎりますね。これは、FOR～NEXTにしかけがあります。37から0へと、1つずつ減っているのがわかるでしょう。STEP －1とマイナス符号がついていますね。1つずつマイナスするということです。そうすることで、CURSORの位置（X方向）が右（37）から左（0）へ変化していくのです。

3 2つのキャラクタを同時に動かす

今までは、1つのキャラクタだけを動かしました。ところが、2つのキャラクタを同時に動かすことも、もちろんできます。

＊2・3　インベーダーとボールが平行に動く

```
100 REM *** 2.3 ***
110 REM *** ナカヨク ナランデ゛ ウコ゛クヨ ***
120 CLS
130 FOR I=0 TO 36
140 CURSOR I,10
150 PRINT" ●";
160 CURSOR I,15
170 PRINT" ♠";
180 NEXT I
190 GOTO 120
```

これを実行すると、"●"と"♠"という2つのキャラクタが上下に並(なら)んで、同時に画面をよこぎります。これも、CURSORのY（タテ）の位置をずらして動かせばいいのです。ここでは5行分ずらして10と15にしてみました。

4 ナナメに動く

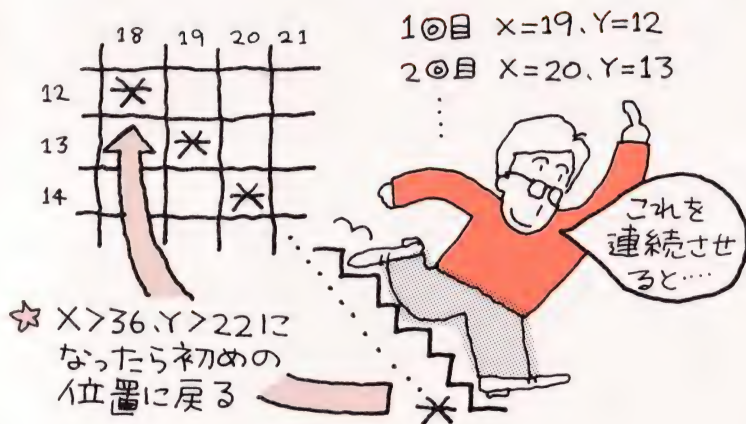
今までは、キャラクタが画面をまっすぐよこぎるだけでした。ここでは、主にナナメに動く動き方を考えてみましょう。

少し長いプログラムになりましたが、次のプログラムを入力して実行してみてください。

＊2・4　流れ星が落ちていく

```
100 REM ***  2.4 ***
110 REM *** ナガレボシ ガ オチテイク ***
120 CLS
130 X=19:Y=12
140 GOTO 190
150 CURSOR X,Y
160 PRINT " ";
170 X=X+1
180 Y=Y+1
190 CURSOR X,Y
200 PRINT"*";
210 IF (X<1) OR (X>36) THEN 240
220 IF (Y<1) OR (Y>22) THEN 240
230 GOTO 150
240 GOTO 120
```

実行すると、"＊"がナナメ右下に動きます。このプログラムでは、CUR-SORのX(ヨコ)とY(タテ)の位置を、変数X、Yにおきかえて変化させています。

また、行番号160で、前に表示した"＊"をスペースで消しています。行番号210、220は、キャラクタが画面の端(はし)にきたときに、また初めからくり返すように判断させています。

前のページのプログラムの行番号170と180を、次のように変えるだけで、キャラクタはナナメ左上に動きます。なぜ左上なのか？　それは170と180にある－1という数字です。これは1つずつ減らしていく、つまりX(ヨコ)方向だったら左へ、Y(タテ)方向だったら上へということですから、結局“※”は左上へ移動することになります。

＊2・5　もちろんナナメ左上もバッチリ！

```
100 REM *** 2.5 ***
110 REM *** ドンドン アガッテイク ***
120 CLS
130 X=36:Y=10
140 GOTO 190
150 CURSOR X,Y
160 PRINT " ";
170 X=X-1
180 Y=Y-1
190 CURSOR X,Y
200 PRINT"*";
210 IF (X<1) OR (X>36) THEN 240
220 IF (Y<1) OR (Y>22) THEN 240
230 GOTO 150
240 GOTO 120
```

5 1つ飛びにスキップ!

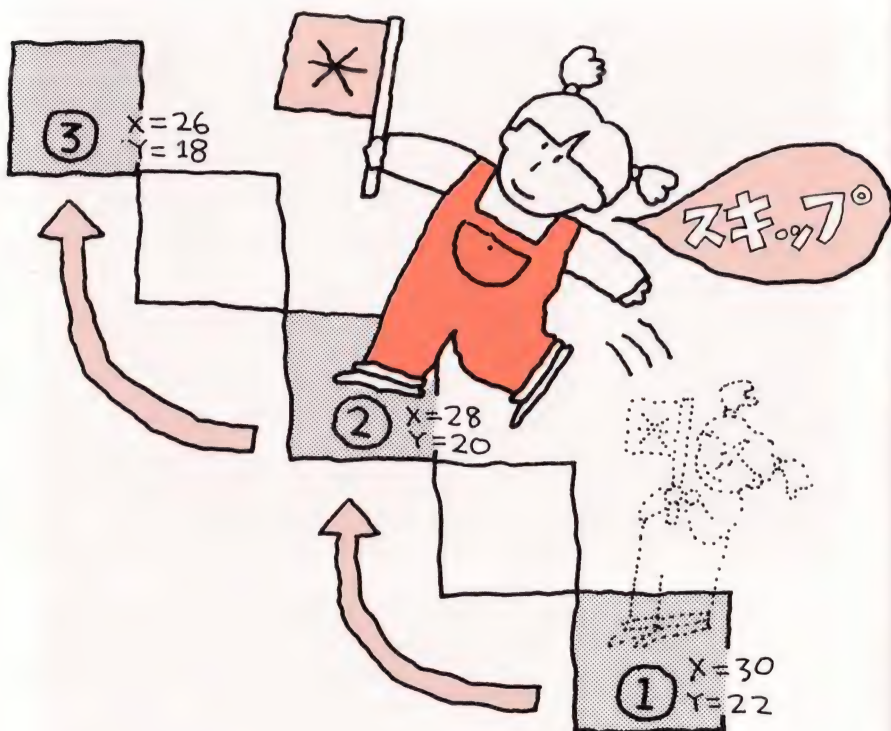
前ページのプログラム中の行番号170と180のXとYを1つずつふやしたり減らしたりするだけでなく、次のように2つずつ減らしてみるとどうなるでしょうか。

＊2・6　スキップをさせたかったら

```
100 REM *** 2.6 ***
110 REM *** ｽｷｯﾌﾟ ｼﾖｳ ***
120 CLS
130 X=30:Y=22
140 GOTO 200
150 CURSOR X,Y:PRINT " ";
160 X=X-2
170 Y=Y-2
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT "*";
210 FOR T=1 TO 40:NEXT T
220 GOTO 150
230 GOTO 120
```

スピードアップ!

１つ飛びに、ナナメ上に動いていくはずです。いってみれば、スキップしながら階段(かいだん)を上がっていく感じです。

大きく動いているので、そのぶん速くなってしまいます。そこで行番号210でカウントを入れることにより、時間かせぎをして調整をしています。これは、次の位置へ移動するたびに40ずつ数えてから行きなさい、という意味で、何かスゴロクと似ていますね。

このように、歩幅を大きくすると、速さもグーンと違ってきます。このプログラムの行番号160と170の「歩幅」を変えていけば、いくらでも速くなります。

6 角度を変えて動かす

もちろん、角度を変えて動かすこともできます。

＊2・7　急角度、ゆるい角度も自由自在

```
100 REM *** 2.7 ***
110 REM *** ﾕﾙｲ ｻｶ ｦ ﾉﾎﾞｯﾃｲｸﾖ ***
120 CLS
130 X=1:Y=22
140 GOTO 200
150 CURSOR X,Y:PRINT " ";
160 X=X+3
170 Y=Y-1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT "*";
210 FOR T=1 TO 40:NEXT T
220 GOTO 150
230 GOTO 120
```

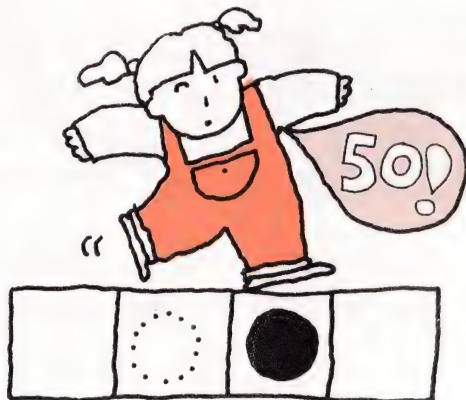
上のプログラムを実行すると、キャラクタはX方向に3、Y方向に－1ずつ動きます。行番号160と170の3と－1を、他の数字にしてみましょう。角度を変えて動かすことができます。Xの値を一定にすれば、タテに動きます。これで、タテ、ヨコ、ナナメ自由自在に動かせますね。

7 速さを変えて動かす

キャラクタが、ただ行ったり来たりしているだけでは面白くありませんね。ここでは、キャラクタの動く速さを変えてみます。まずは、スローモーションから。

＊2・8　ユックリ走らせたいなら

```
100 REM *** 2.8 ***
110 REM *** ユックリ ハシロウ ***
120 CLS
130 FOR I=0 TO 36
140 CURSOR I,15
150 PRINT" ●";
160 FOR T=1 TO 50:NEXT
170 NEXT I
180 GOTO 120
```

このプログラムを実行すると、"●"の動き方がさっきよりゆっくりになります。これは、行番号160のFOR～NEXTで時間かせぎをしているのです。

ここでは、変数Tが1～50までになっていますが、50のところをもっと大きな数字にすると、より遅くなります。逆に、小さな数字にすると速くなり、行番号160をとってしまうと、一番速くなります。それより速くするには、すでに述べたように歩幅を大きくする方法があります。

数字を変えて、いろいろためしてみましょう。

行番号160を次のように変えてみましょう。

＊2・9　だんだん遅くなってきた……

```
100 REM *** 2.9 ***
110 REM *** ﾀﾞﾝﾀﾞﾝ ｵｿｸ ﾅﾙﾖ ***
120 CLS
130 FOR I=0 TO 36
140 CURSOR I,15
150 PRINT" ●";
160 FOR T=1 TO I*2:NEXT
170 NEXT I
180 GOTO 120
```

＊2・10　だんだん速くなってきた……

```
100 REM *** 2.10 ***
110 REM *** ﾀﾞﾝﾀﾞﾝ ﾊﾔｸ ﾅﾙﾖ ***
120 CLS
130 FOR I=0 TO 36
140 CURSOR I,15
150 PRINT" ●";
160 FOR T=1 TO 72-I*2:NEXT T
170 NEXT I
180 GOTO 120
```

プログラム2・9では、“●”の動きがだんだん遅くなり、プログラム2・10ではだんだん速くなっていきます。これは、1回ごとに待つ時間をふやしていったりへらしていったりしているからです。

8 足跡(あしあと)を残して動く

タテ、ヨコ、ナナメの動き方は、これでわかったと思います。この動き方の基本におまけをつけて、もう少し面白くしてみましょう。

＊2・11　移動の足跡を残したいのなら

```
100 REM *** 2.11 ***
110 REM *** ｱｼｱﾄｶﾞ ﾃﾝﾃﾝﾃﾝ･･･ ***
120 CLS
130 X=37:Y=0
140 GOTO 200
150 CURSOR X,Y:PRINT ".";
160 X=X-3
170 Y=Y+1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT "*";
210 FOR T=1 TO 40:NEXT T
220 GOTO 150
230 GOTO 120
```

上のプログラムは、"＊"の動いたあとに"．"の足跡を残していきます。これは、行番号150で今まではスペースを使って"＊"を消していましたが、代わりに"．"を表示するようにしただけです。

3

キーでキャラクタを操作する

止めたり、動かしたり…キミだけの暗号をつくっちゃおう！

スイッチオン！ する前に

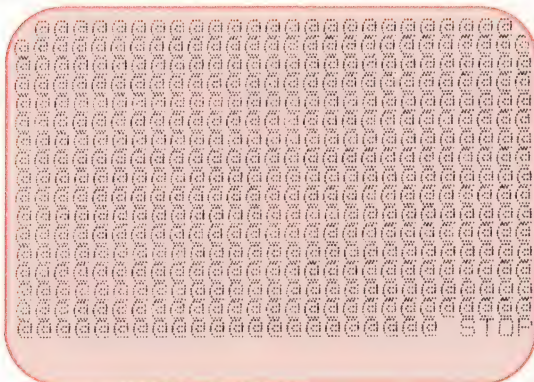
画面上を、キャラクタが勝手に動きまわるだけでは、面白いゲームはつくれません。やはり、動かしたり、止めたり、方向を変えたりをキーひとつで操作したいものです。そこで、次はキャラクタの動きを、キーで操作する方法を紹介しますね。

1 キーでストップさせる

キーひとつでキャラクタの操作をするには、INKEY$（インキーダラー）を使います。まずは、動いているものを止めてしまう、簡単なプログラムから見てみましょう。

＊3・1　INKEY$を使って止めちゃおう

```
100 REM *** 3.1 ***
110 REM *** ｲﾂ ｽﾄｯﾌﾟ ｻｾﾙﾉｶﾅ? ***
120 CLS
130 FOR I=1 TO 30:NEXT I
140 PRINT"@";
150 A$=INKEY$
160 IF A$="" THEN 140
170 PRINT " STOP"
```

実行すると、画面には次から次へと“@”を表示していきます。ほおっておくと、いつまでも@を表示し続けます。ここで、どのキーでもかまいませんから、キーを１つ押します。キーを押すと同時に“STOP”を表示して止まります。キーを押したかどうかを判断しているのは、行番号**150**のINKEY$と**160**のIF～THENです。

はやく止めないと画面がいっぱいになるヨ！

2 キーでスタート!

次は、キーを押すとスタートするというプログラムです。

＊3・2　どれか1つのキーを押して……

```
100 REM *** 3.2 ***
110 REM *** ハシリダ゛スヨヨヨヨヨ ***
120 CLS
130 X=0:Y=0
140 GOTO 200
150 CURSOR X,Y:PRINT" "
160 X=X+2
170 Y=Y+1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT"*";
210 FOR T=1 TO 20:NEXT T
220 GOTO 150
230 A$=INKEY$
240 IF A$="" THEN 230
250 GOTO 120
```

上のプログラムは、実行しただけでは何も起こりません。どれかのキーを押してはじめて"＊"が飛び出すしかけになっています。その秘密(ひみつ)は、行番号**230**と**240**にあります。つまり、INKEY$が働くと同時に、"＊"がナナメ右下30度の角度に動いていくというものです。

このようなINKEY$の働きを生かして、キーを押したらゲームがスタートするというようにしてみたらどうでしょう。画面はのせていません。あなた自身で確かめてみてください。これは、便利に使えそうですね。

3 使うキーを決める

しかし、どのキーでも押せばいいというのではつまらないかもしれません。どうせなら、Sはスタート、Tはストップというふうに、キーを決めてしまいたいものです。そこで、特定のキーを使って操作する方法を考えてみましょう。

＊3・3　Sはスタートに決めちゃえ

```
100 REM *** 3.3 ***
110 REM *** ｷｰ ｦ ｵｽﾄ ｳｺﾞｸﾖ ***
120 CLS
130 X=0:Y=0
140 GOTO 200
150 CURSOR X,Y:PRINT" "
160 X=X+2
170 Y=Y+1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT"*";
210 FOR T=1 TO 20:NEXT T
220 GOTO 150
230 A$=INKEY$
240 IF A$<>"S" THEN 230
250 GOTO 120
```

今度は、Sを押したときだけ"＊"が飛び出します。他のキーを押しても、ウンともスンともいいません。これは、行番号**240**でS以外のキーが押されたら、もう一度**230**に戻って正しいキーを押しなさい、というものです。Sが押されてはじめて、行番号**120**に進み、動き始めるのです。

4 特殊キーを使うとらしくなる

文字キーのほかに、←→↑↓や CR などの特殊キーを使ってキャラクタを操作すると、かなりそれらしくなります。

★←と→を使う

まずは、←→を使って、左右に移動するプログラムを見てみましょう。

＊3・4　矢印の向きがそのまま移動する方向になる

```
100 REM *** 3.4 ***
110 REM *** ドッチ ニ イクノカナ? ***
120 CLS
130 X=0:Y=12
140 GOTO 200
150 CURSOR X,Y:PRINT" "
160 X=X+DX
170 IF X<0 THEN X=0
180 IF X>37 THEN X=37
190 CURSOR X,Y:PRINT"*";
200 A$=INKEY$
210 IF A$=CHR$(28) THEN DX=1
220 IF A$=CHR$(29) THEN DX=-1
230 GOTO 150
```

プログラムを実行すると、“＊”を左端に表示します。→を押すと、“＊”は右に移動します。反対に、←を押すと左に移動します。

もっとも特殊キーは、文字キーなどのように"(ダブルクォーテーション)で囲んで指定することができません。“←”とか“↑”には、できないのです。そこで、キャラクタコードを使います。特殊キーにも、それぞれキャ

ラクタコードがあります。たとえば、行番号**210**で指定している(28)のキャラクタコードをもつキーは、→です。行番号**220**で指定している(29)は、←のキャラクタコードです。

ちなみに、その他の特殊キーのキャラクタコードは、次のようになっています。

●特殊キーのキャラクタコードはこうなっている

キャラクタコード	機　　　　能
1	NULL　文字なし
——	BREAK　プログラムの実行中止
5	カーソル以降の文字をクリア
7	BELL　ピッと音を出す
8	DEL　文字をデリート
9	HT　水平　TAB
10	LF　ラインフィード
11	HM　カーソルを左上端に戻す
12	CLR　画面のクリア
13	CR　キャリッジリターン
14	カナ↔英数字切りかえ
15	↘↗画面をテキスト↔グラフィック切りかえ
16	標準文字サイズ
17	横2倍文字サイズ（SCREEN 2）
18	INS　インサート
19	キー入力（A～Z）シフトなし大文字
20	〃　（a～z）シフトなし小文字
21	ラインをクリアしカーソルを左端に戻す
22	ノーマルモード
23	GRAPH　キー入力グラフモード↔英字切換
24	クリック音のON↔OFF切換
28	⇨　カーソル移動
29	⇦　〃
30	⇧　〃
31	⇩　〃

前のページのプログラムに、次のように行番号**205**を加えてください。

＊3・5　キーを押さないと動かない

```
100 REM *** 3.5 ***
110 REM *** ｷｰ ｦ ｵｻﾅｲﾄ ｳｺﾞｶﾅｲﾖ ***
120 CLS
130 X=0:Y=12
140 GOTO 200
150 CURSOR X,Y:PRINT" "
160 X=X+DX
170 IF X<0 THEN X=0
180 IF X>37 THEN X=37
190 CURSOR X,Y:PRINT"*";
200 A$=INKEY$
205 IF A$="" THEN 200
210 IF A$=CHR$(28) THEN DX=1
220 IF A$=CHR$(29) THEN DX=-1
230 GOTO 150
```

上のプログラムを実行すると、キーを押していないとき(＝″　″)は″※″が動かなくなります(A$＝INKEY$)。行番号の**200**と**205**の間を行ったり来たりということになるからです。

また、反対にキーを押しっぱなしだと″※″が動かないというふうにもできます。上のプログラムの行番号**130**と**205**を変えて、行番号**206**をつ

けたしてみましょう。次のプログラムがそれです。

＊3・6　一部を変えてみよう

```
100 REM *** 3.6 ***
110 REM *** ﾄﾞﾉ ｷｰ ﾆ ﾇﾙﾉｶﾅ? ***
120 CLS
130 X=0:Y=12:B$=""
140 GOTO 200
          :
          :
200 A$=INKEY$
205 IF A$=B$ THEN 200
206 B$=A$
210 IF A$=CHR$(28) THEN DX=1
```

上のプログラムは、キーを押しっぱなしだと"＊"は動きません。"＊"を動かすには、キーを１回ずつ押してください。どうしてこうなるのかというと、A$には押したキー、B$には前に押したキーが入っているからなのです。A$とB$の関係は次のようになっています。

キーの状態	始め	押す →	押しっぱなし →	はなす	押していない	押す ←
A$の中	ヌル	→	→	ヌル	ヌル	←
B$の中	ヌル	ヌル	→	→	ヌル	ヌル
		↓ １つ右に動く				↓ １つ左に動く

★←→↑↓で上下左右に動かす

←→で左右に動かすプログラムを、もう少し改良して、↑↓で上下にも動くようにしてみましょう。

＊3・7　上下左右に動くようにしたい……

```
100 REM *** 3.7 ***
110 REM *** イロイロナ ホウ ヘ ウゴカシタイナ ***
120 CLS
130 X=19:Y=12
140 GOTO 210
150 CURSOR X,Y:PRINT" "
160 X=X+DX:Y=Y+DY
170 IF X<0 THEN X=0
180 IF X>37 THEN X=37
190 IF Y<0 THEN Y=0
200 IF Y>23 THEN Y=23
210 CURSOR X,Y:PRINT"*";
220 A$=INKEY$
230 IF A$="" THEN 220
240 IF A$=CHR$(28) THEN DX=1:DY=0
250 IF A$=CHR$(29) THEN DX=-1:DY=0
260 IF A$=CHR$(30) THEN DY=-1:DX=0
270 IF A$=CHR$(31) THEN DY=1:DX=0
280 GOTO 150
```

上のプログラムを実行すると、↓↑←→を使って、"＊"を上下左右に動かすことができます。行番号**240**～**270**で、次のような処理をしています。

キー	処　理	キー	処　理
→	右に1つ動かす	↓	下に1つ動かす
←	左に1つ動かす	↑	上に1つ動かす

ところで、上のプログラムの行番号**150**にスペースを指定してありますが、代わりに"■"を入れてみましょう。

＊3・8　簡単な絵が描ける

```
100 REM *** 3.8 ***
110 REM *** アレレレ エ ガ カケタ! ***
120 CLS
130 X=19:Y=12
140 GOTO 210
150 CURSOR X,Y:PRINT"■"
160 X=X+DX:Y=Y+DY
170 IF X<0 THEN X=0
180 IF X>37 THEN X=37
190 IF Y<0 THEN Y=0
200 IF Y>23 THEN Y=23
210 CURSOR X,Y:PRINT"*";
220 A$=INKEY$
230 IF A$="" THEN 220
240 IF A$=CHR$(28) THEN DX=1:DY=0
250 IF A$=CHR$(29) THEN DX=-1:DY=0
260 IF A$=CHR$(30) THEN DY=-1:DX=0
270 IF A$=CHR$(31) THEN DY=1:DX=0
280 GOTO 150
```

これでも
立派な絵だヨ！

プログラムを実行すると、↑↓←→で"＊"を動かして簡単な絵が描けるようになります。これは足跡(あしあと)を残す働き(40ページ参照)と同じ原理です。

★↑↓←→を使ってキャラクタを動かす

もう1つ、↑↓←→を使ってキャラクタを動かすプログラムをつくってみました。これも長いプログラムですが、辛抱(しんぼう)して入力してみてください。きっと面白い画面が現われるはずです。

＊3・9　こんな面白い画面が……

```
5 REM *** 3.9 ***
8 REM *** ムムムム ナ ナンダ? ***
10 SCREEN 2,2:CLS
20 X=128:Y=96
30 CURSOR X,Y:PRINT"A"
50 Z$=INKEY$
60 IF Z$=CHR$(28) THEN E=8:GOSUB 200
65 IF Z$=CHR$(29) THEN E=-8:GOSUB 200
70 IF Z$=CHR$(30) THEN E=-8:GOSUB 300
75 IF Z$=CHR$(31) THEN E=8:GOSUB 300
80 IF Z$= "" THEN 50
90 GOTO 50
200 REM    <----------->
210 CURSOR X,Y:PRINTCHR$(8)
220 X=X+E:IFX<=0 OR X>=256 THEN X=X-E
230 CURSOR X,Y:PRINT"A"
290 RETURN
300 REM    ^-----------v
310 CURSOR X,Y:PRINTCHR$(8)
320 Y=Y+E:IFY<=0 OR Y>=180 THEN Y=Y-E
330 CURSOR X,Y:PRINT"A"
390 RETURN
```

プログラムを実行すると、↑↓←→を使って"A"を上下左右に動かすことができます。

今度は、キャラクタがちょっと変わった動き方をするプログラムを紹介します。

＊3・10　キャラクタが変わった動きをする

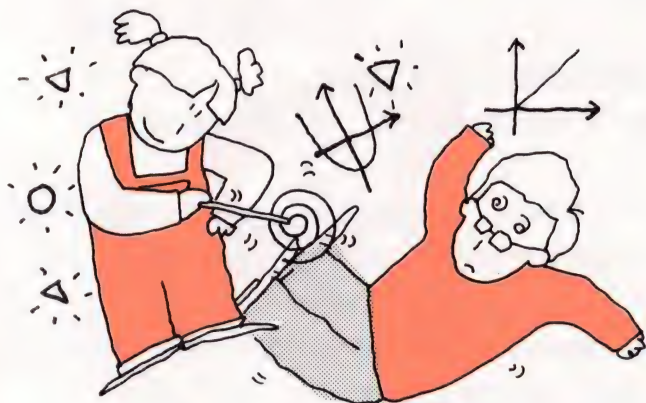
```
6 REM *** 3.10 ***
7 REM *** ヒョーキン エイリアン ダ゛ー ***
10 CLS:X=20:Y=15
20 CURSOR X,Y:PRINT " "
30 Z$=INKEY$
40 IF Z$=CHR$(28) THEN E=1:GOSUB 100
50 IF Z$=CHR$(29) THEN E=-1:GOSUB 100
60 IF Z$=CHR$(12) THEN E=-2:GOSUB 200
70 IF Z$=CHR$(8) THEN E=-6:GOSUB 200
80 GOTO 30
100 CURSORX,Y:PRINT"♣"
110 FOR N=0 TO 10:NEXT N
115 X=X+E
120 CURSORX,Y:PRINT"♣"
130 RETURN
200 CURSORX,Y:PRINT"♣"
210 Y=Y+E
220 CURSORX,Y:PRINT"♣"
230 FOR N=0 TO 50:NEXT
240 CURSORX,Y:PRINT"♣"
250 Y=Y-E
260 CURSORX,Y:PRINT"♣"
280 RETURN
```

プログラムを実行すると、←→を使ってUFOを左右に動かすことができます。さらに、CTRL を押しながらLを押すと、UFOが小さくジャンプし、CTRL を押しながらHを押すと、大きくジャンプします。

4

CGにアタック！

(コンピュータ・グラフィック)

ゲームのキャラクタ、背景づくりのテクニックをつかもう!!

スイッチオン！ する前に

ここでは、画面上にこまかい点を打つことによって図形を描き(か)ます。ゲームには、キャラクタだけでなく、背景もあります。ここで紹介するグラフィックのテクニックを使って、きめのこまかい絵を描いてみましょう。

1 もう1つの画面—グラフィック画面

「お父さん、コンピュータでお絵描きができるって聞いたけど、ホントにできるの?」

「ウン、できるよ。でもネ、今まで説明してきたものとは別の画面を使わなくちゃいけないんだ」

「フーン、でも別の画面っていうけど、テレビには1つしか画面がないじゃない」

「ここがマイコンのいいところなんだ。要するに、切換えスイッチ1つで別の画面が登場するってワケなのさ」

「へえー、よくわからないけど、何だか楽しそうだね」……

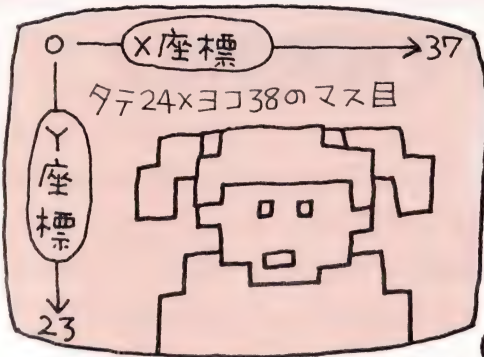
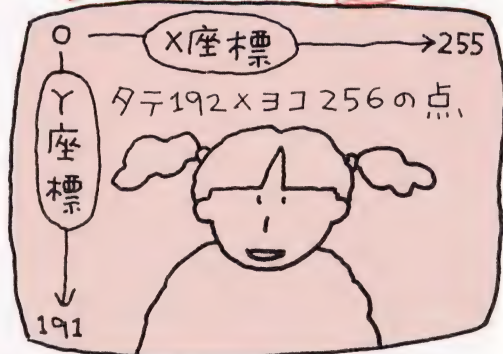
そうです、SEGA SK-1100には実は、2つの画面があるんです。それはテキスト画面とグラフィック画面の2つです。今までのプログラムで使っていたのは、このうちのテキスト画面だけでした。

テキスト画面については、29ページで簡単に説明しましたが、ここでもう一度、画面について考えてみましょう。

物事をするにはとにかく初めが大切だといいますが、画面でも元の位置、スタート地点があります。画面の左上の隅がそうです。ここを原点といい数字で 0、0、の位置をいいます。よく、ゼロからの出発、なんていいますが、マイコンの世界でもこのリクツがまかり通っています。

そして、原点からヨコ方向をX座標、タテ方向をY座標といいます。座標とは「位置づけ」ということです。

テキスト画面だと、これがタテ24、ヨコ38のマス目に分かれており、グラフィック画面では、何とタテ192、ヨコ256の点に分かれます。

次ページに切換えのしかたを図示しました。ためしてみてください。

0
X座標
37
タテ24×ヨコ38のマス目
Y座標
23
☆スイッチONにしたときはテキスト画面
画面切り換え
テキスト画面
● SHIFT + BREAK
● CTRL + O
● PRINT CHR$(15)
どれかを使えばいいのサ
グラフィック画面
0
X座標
255
タテ192×ヨコ256の点
Y座標
191
☆画面が白くなるヨ

2 グラフィック画面を使う

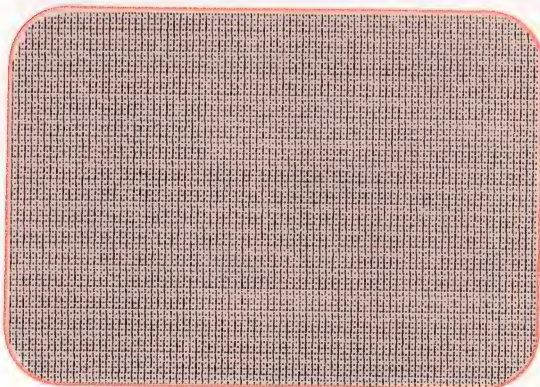
では、論よりRUN、実際にプログラムを動かして、グラフィック画面を使ってみましょう。

プログラム中で、テキスト画面からグラフィック画面に変えるには、SCREEN(スクリーン)を使います。次のプログラムを入力してください。

＊4・1　アミの目もようを描く

```
100 REM *** 4.1 ***
110 REM *** ｱﾐﾉﾒ ｶﾞ ｲｯﾊﾟｲ ﾃﾞﾃｸﾙﾖ ***
120 SCREEN 2,2:CLS
130 FOR X=0 TO 255 STEP 2
140 LINE (X,0)-(X,191)
150 NEXT X
160 FOR Y=0 TO 191 STEP 2
170 LINE (0,Y)-(255,Y)
180 NEXT Y
190 GOTO 190
```

実行すると、画面にアミの目もようを描いていきます。

ワッ！

ここでアミの目もようの線を引いているのは、LINEです。行番号140のLINEがタテの線、行番号170のLINE がヨコの線を引いています。そして、それぞれの線をX座標は0から255まで、Y座標は0から191まで、FOR～NEXTで繰り返して、アミの目もようをつくっています。つまり、画面いっぱいにアミの目がかかるようになっているわけです。

行番号120に、グラフィック画面を表示するSCREENがあります。

このSCREENにちょっと手を加えてみます。

行番号120を

```
SCREEN 2,1
```

に書き直し、行番号185として次の行をプログラムにつけ加えましょう。

```
185 SCREEN 2,2
```

さて、ここでもう一度プログラムを実行します……。

RUNと入力しても、しばらくは何もしません。少し待っていると、突然画面が変わって、さきほどのアミの目もようが現われます。

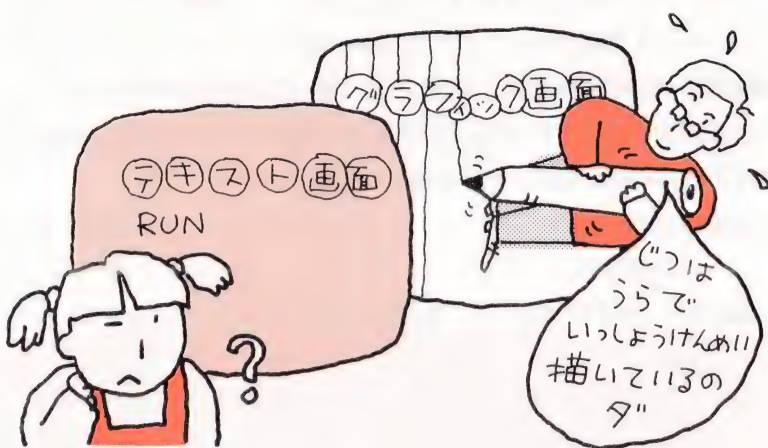
タネあかしをしましょう。秘密はSCREENにあります。SCREENは、次のような指定になっているのです。

SCREEN　表示画面，書き込み画面

1 ： テキスト画面

2 ： グラフィック画面

つまり、行番号120の SCREEN 2，1 は、テキスト画面を表示しながら、グラフィック画面にアミの目もようを書き込んでいるということになります。

描き終わったところでグラフィック画面を表示します（行番号185）から、描いているところを見せないで、画面がパッと変わったように見せることができるのです。

ところで、グラフィック画面は、実行が終わると自動的にテキスト画面に戻ってしまいますので、行番号190のようにプログラムを終わらせないようにすることも忘れないようにしましょう。

似たようなプログラムをもう１つのせておきます。

＊４・２　画面いっぱいに箱を描く

```
100 REM *** 4.2 ***
110 REM *** ハコ ガ゛ イッハ゜イ テ゛テクルヨ ***
120 SCREEN 2,2:CLS
130 FOR X=0 TO 240 STEP 20
140 FOR Y=10 TO 170 STEP 20
150 LINE (X,Y)-(X+15,Y+15),,B
160 NEXT Y
170 NEXT X
180 GOTO 180
```

実行すると画面いっぱいに箱を描きます。ためしてみてください。

3 こまかい点を打つ

テキスト画面は、タテ24、ヨコ38のマス目からできているといいましたが、このマス目の１つ１つは点（ドット）がタテヨコ８つずつ、つまり64のドットからできています。その64の点をいっぺんに指定しますから、目があらくなります。

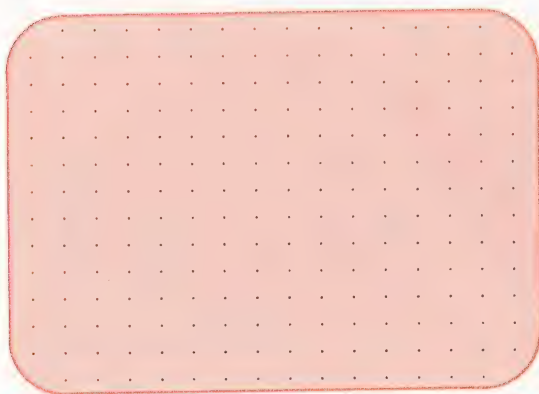
その点、グラフィック画面は１つのドットずつに指定できますから、きめのこまかい絵が描けることになるのです。

ではでは――。グラフィック画面の優秀さを、ここでちょっと披露(ひろう)してみましょう。エヘン！

簡単なところで、画面に点を打ってみます。次のプログラムを入力してみてください。

＊4・3　点を打ってみよう

```
100 REM *** 4.3 ***
110 REM *** テンテンテン ***
120 SCREEN 2,2:CLS
130 FOR X=0 TO 240 STEP 10
140 FOR Y=0 TO 190 STEP 10
150 PSET (X,Y)
160 NEXT Y
170 NEXT X
180 GOTO 180
```

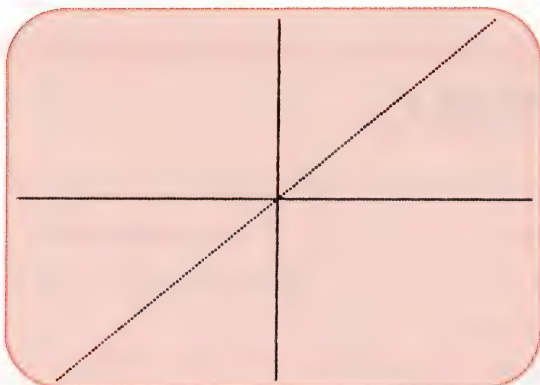
このプログラムを実行すると、下の画面のように、点がいっぱいに広がっていきます。グラフィック画面ならでは、ですね。なお、1つの点が1ドットです、念のため。

何に使えるかな？

4 グラフを描く

画面の座標は左上が原点(0,0)でした。この原点を画面のまん中、つまり座標でいうと(128,96)にもってきてグラフを描くプログラムをつくってみました。それには、POSITION(ポジション)を使います。

＊4・4 座標軸を描き, グラフをつくる

```
100 REM *** 4.4 ***
110 REM *** ｸﾞﾗﾌ ｶﾞ ｶｹﾙﾖ ***
120 SCREEN 2,2:CLS
130 COLOR 7
140 POSITION (128,96),0,1
150 LINE (-120,0)-(120,0)
160 LINE (0,90)-(0,-90)
170 COLOR 6
180 FOR X=-120 TO 120
190 Y=X
200 IF ABS(Y)>90 THEN 220
210 PSET (X,Y)
220 NEXT X
230 GOTO 230
```

このプログラムをRUNさせると、まず初めに下のような座標軸を描きます。そして、そのあとY=Xのグラフを描くことになります。

POSITIONの指定は、次のようになっています。

POSITION（X軸の0（ゼロ），Y軸の0）Xの軸方向，Yの軸方向

0：Xは右に増加，Yは下に増加
1：Xは左に増加，Yは上に増加

ところで、このプログラムの行番号190の式を変えるだけで、いろいろなグラフを描くことができます。たとえば、こんなふうに……。

```
190 Y=X+30
```

Y=Xと平行な(0,30)を通る直線。

```
190 Y=X*X/100
```

放物線。100のところを大きい数にするほど小さい単位の放物線が描けます。

```
190 Y=SIN(PI*X/
60)*90
```

サイン曲線。

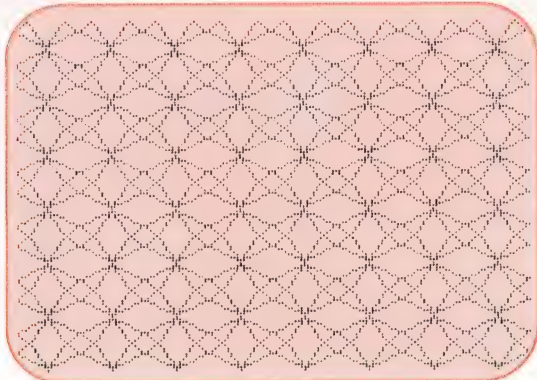
学校で習った公式を思いだしていろいろためしてみましょう。

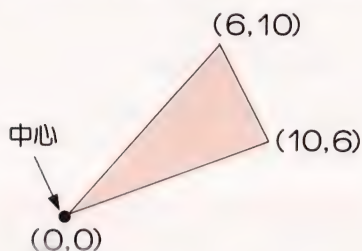
5 連続もようを描く

POSITIONのもつ、原点の位置を変えたり軸の方向を変えるという働きを利用して、連続もようのプログラムをつくってみました。

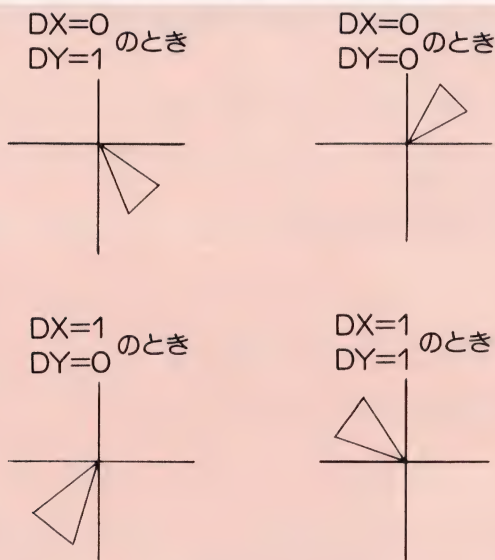
＊4・5 画面いっぱいに花を描こう

```
100 REM *** 4.5 ***
110 REM *** オナジ゛ モヨウ ヲ クリカエスヨ ***
120 SCREEN 2,2:CLS
130 FOR X=10 TO 230 STEP 20
140 FOR Y=10 TO 170 STEP 20
150 FOR DX=0 TO 1
160 FOR DY=0 TO 1
170 POSITION (X,Y),DX,DY
180 LINE (0,0)-(6,10)
190 LINE-(10,6):LINE-(0,0)
200 NEXT DY
210 NEXT DX
220 NEXT Y
230 NEXT X
240 GOTO 240
```

このプログラムを実行すると画面いっぱいに花を描いていきます。

さて、プログラムの中で、実際に描くという処理をしているのは行番号180と190だけです。

行番号180と190では、次のような三角形の花びらを描いています。

POSITIONを使うことによって、この花びらの中心と向きをいろいろ変化させています。

行番号170を見てください。POSITIONの変数XとYは中心を、変数DXとDYは向きを変える働きをしています。

●DXとDYが0(ゼロ)か1だったら

6 色をぬる

三角形や四角形、円……いろいろな形をつくっても、ただ線と面だけでは物足りません。三角屋根は赤、窓は青、山は緑……というように、やはり色をぬってこそ、グラフィックだ、という気がしますね。

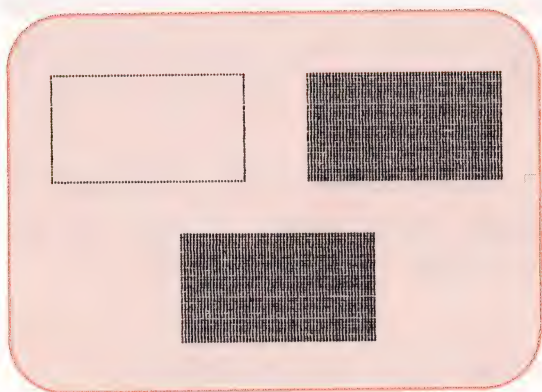
このときに、すごい力を発揮するのが、LINE(ライン)とPAINT(ペイント)です。

ただし、LINEには必ずBF（Box Fill(ボックス・フィル)　箱の中をぬりつぶす）というものをつけなければなりません。しかも、LINEが通用するのは、四角形だけなのです。

その点、PAINTは、四角形はもちろん、三角形、円、その他の形にもできますから、大変便利です。PAINTは、囲まれた部分をぬるという働きをします。ところが、この囲んだ線が途中でちょっとでもとぎれていると、色がはみ出してしまいます。ご用心、ご用心。

＊4・6　赤、青、黒……いろいろな色でぬる

```
100 REM *** 4.6 ***
110 REM *** ハコ ニ イロ ヲ ヌロウ ***
120 SCREEN 2,2:CLS
130 LINE(40,20)-(100,60),1,B
140 LINE(120,20)-(180,60),5,BF
150 LINE(80,80)-(140,120),8,B
160 PAINT(81,81),8
170 GOTO 170
```

では、このプログラムを実行してみましょう。

上が画面ですが、これも実際は赤と青、黒の３つの色がついています。
行番号130の１、140の５、150の８などが、色の番号なのです。

なお、色は次の数字で表わします。

0	透明	9	うすい赤
1	黒	10	こい黄
2	緑	11	うすい黄
3	うすい緑	12	こい緑
4	こい青	13	マゼンタ
5	うすい青	14	灰色
6	こい赤	15	白
7	水色		
8	赤		

7 ぬったり、消したり

絵の具やクレヨンでは一度ぬりつぶしたものの上に、ちがう色をぬり重ねることができますが、コンピュータではそれができません。このようなときには前にぬってあった色をいったん消してから、もう一度ぬるというやり方をします。これには、BLINE（ビーライン）やPRESET（プリセット）（画面上のドットを消す）を使います。

では、次のプログラムをためしてみましょう。

＊4・7　一度ぬったものを消して新しい色でぬる

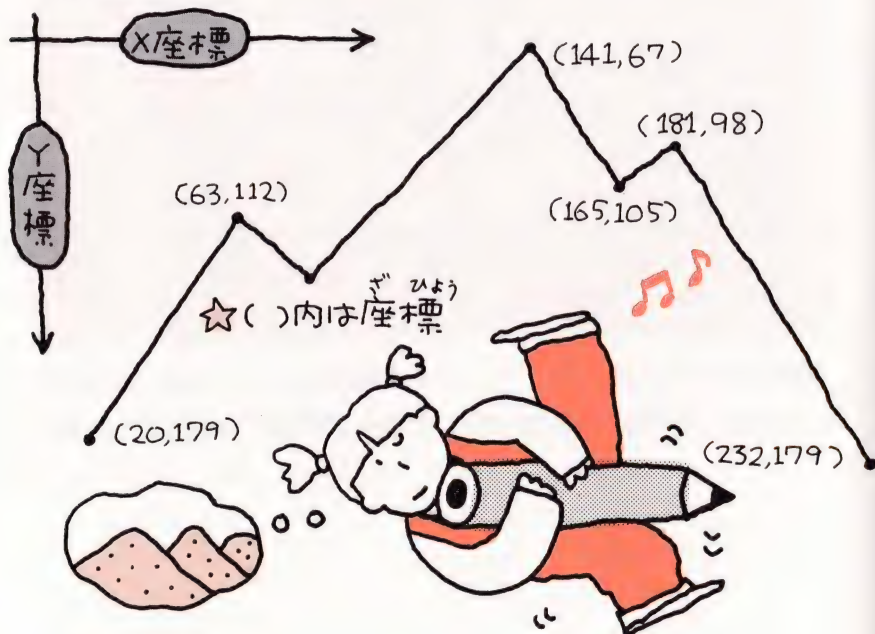
```
100 REM *** 4.7 ***
110 REM *** ﾁｶﾞｳ ｲﾛ ﾆﾓ ﾃﾞｷﾙﾖ ***
120 SCREEN 2,2:CLS
130 LINE (20,20)-(235,171),13,BF
140 BLINE (40,40)-(215,151),,BF
150 FOR X=20 TO 230 STEP 10
160 FOR Y=20 TO 170 STEP 10
170 PRESET (X,Y)
180 NEXT Y
190 NEXT X
200 LINE (60,60)-(195,131),7,BF
210 BLINE (80,80)-(175,111),,BF
220 GOTO 220
```

画面はのせませんが、このプログラムを実行すると、一度ぬったものを消してから、次の色をぬるというコンピュータ独特の方法がわかりますよ。

8 山を描く

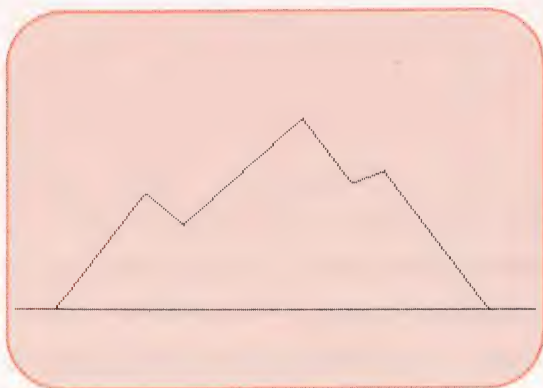
線の引き方として、LINEを使って点と点を結んだり、それをFOR～NEXTでくり返して、アミの目もようを描いたりする方法をみてきました。

ところで、線でも不規則に折れ曲がっている場合などは、どうしたらいいのでしょう。たくさんのLINEを並べて点と点を結んでいったらたいへんです。そこでREAD(リード)～DATA(データ)を使ってラクラクと描いてしまう方法を紹介します。なお、READとは、DATA文のデータを読みとるという意味をもっています。

もしあなたが、山の絵を描きたいとしたら、まず文房具屋さんで方眼紙(ほうがんし)を買ってきてください。そして、下の図のように線を書き、その座標(ざひょう)をメモしておきます。

＊4・8 山の稜線を書く

```
100 REM *** 4.8 ***
110 REM *** ｴﾍﾞﾚｽﾄ ｶﾅ ﾏｯﾀｰﾎﾙﾝ ｶﾅ? ***
120 SCREEN 2,2:CLS
130 COLOR 9
140 LINE (0,180)-(255,180),13,BF
150 READ X,Y
160 PSET (X,Y),2
170 READ X,Y
180 IF X<0 THEN GOTO 180
190 LINE -(X,Y),2
200 GOTO 170
210 DATA 20,179,63,112,82,130,141,67
220 DATA 165,105,181,98,232,179,-1,-1
```

上のプログラムを実行すると、下の画面のように山を描きます。

小さい山、大きい山、けわしい山……

行番号210から結びたい点の座標をX座標、Y座標の順に並べていきます。データの終わりに−1とあるのは、行番号180でデータが終わったかどうかを判断するためです。

ここで、行番号210からのデータ部分を書き変えると、自分の好きな絵にすることができます。マッターホルンでも富士山でも自由自在ですよ！

9 キャラクタを大きくする

グラフィック画面では、キャラクタのヨコの幅を大きくすることができます。次のプログラムを実行してみましょう。

＊4・9 "SEGA"の文字を大きくする

```
100 REM *** 4.9 ***
110 REM *** "セガ" ノ ナマエ ガ デルヨ ***
120 SCREEN 2,2:CLS
130 CURSOR 100,70
140 COLOR 4
150 PRINT"SEGA"
160 PRINT CHR$(17)
170 COLOR 1
180 CURSOR 88,90
190 PRINT"SEGA"
200 PRINT CHR$(16)
210 GOTO 210
```

上のプログラムを実行すると"SEGA"という文字を表示します。よく見ると普通の文字より大きく見えます。なぜでしょうか？　行番号160でPRINT CHR$(17)としているからです。もとの大きさに戻すには17のところを16とします。

なお、テキスト画面では色を変えると全部のキャラクタの色が変わってしまいますが、グラフィック画面ではキャラクタごとに1つずつ色を指定することができます。

SEGA

SEGA

キミの名前でももちろんOK！

10 多角形をつくる

辺の数を入力すると、その数の辺からなる多角形をつくるというプログラムです。

＊4・10　五角形から千角形(？)まで多角形をつくる

```
100 REM *** 4.10 ***
110 REM *** ゴカクケイ ヲ カイテミヨウ ***
120 INPUT "N=?";N
130 SCREEN 2,2:CLS
140 POSITION (80,48),0,1
150 S=2*PI/N:R=47
160 PSET (0,R)
170 FOR I=1 TO N-1
180 X=INT(SIN(I*S)*R+0.5)
190 Y=INT(COS(I*S)*R+0.5)
200 LINE -(X,Y)
210 NEXT I
220 LINE -(0,R)
230 A$=INKEY$:IF A$="" THEN 230
240 GOTO 120
```

上のプログラムを実行すると、"N=?"と、まず辺の数をきいてきます。そこで、好きな数を入力すると、多角形を描きます。5と入力すると五角形を描くという具合です。もし千角形をつくりたいなら1000と入力してもいいわけです。

描き終わってから、どれかひとつ好きなキーを押すともう一度辺の数を入力して多角形を描くことができます。どんな多角形をつくるか、ぜんぶ、あなたの心次第(こころしだい)です。

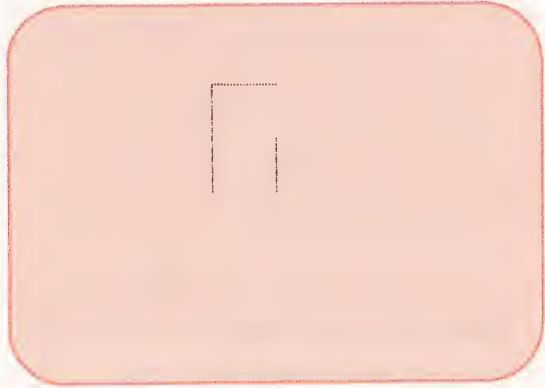
マイコンに命令できるって、気持ちのいいものですね。

11 LOGO(ロゴ)をつくる

BASICと同じようなコンピュータの言葉にLOGO(ロゴ)というものがあります。これは、タートルグラフィックとも呼ばれ、向きや長さを指定するだけで簡単に線が引けるという、ちょっと面白(おもしろ)いものです。本物のLOGOにはもっといろいろ便利な命令がありますが、ここではBASICを使って、LOGOのさわりの部分をつくってみましょう。

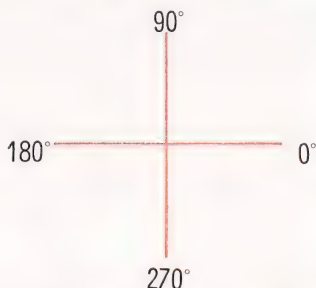
＊4・11　BASICを使ってLOGOの代わりを

```
100 REM *** 4.11 ***
110 REM *** ドンドン ススンデイクヨ ***
120 SCREEN 2,2:CLS
130 POSITION (128,96),0,1
140 C=1:PS=1
150 X0=0:Y0=0
160 RX=90:DX=20:GOSUB 1020
170 PS=0:GOSUB 1020
180 PS=1:RX=180:GOSUB 1020
190 RX=270:DX=40:GOSUB 1020
200 GOTO 200
1000 REM ***FORWARD***
1010 REM ***X0,Y0,RX,DX,C,PS***
1020 RX=RX MOD 360
1030 PP=PI/180
1040 X1=X0:Y1=Y0
1050 X0=X0+COS(RX*PP)*DX
1060 Y0=Y0+SIN(RX*PP)*DX
1070 IF PS=0 THEN RETURN
1080 LINE (X1,Y1)-(X0,Y0),C
1090 RETURN
```

プログラムを実行すると、次のような絵を描きます。

さて、このプログラムのうち、行番号140から190の変数の指定を変えるだけで、自分の好きな絵が描けるようになります。それぞれの変数の意味は、次のとおりです。

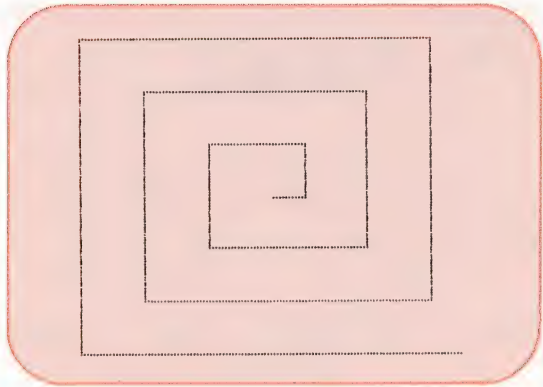
XO	現在のX座標の位置	
YO	現在のY座標の位置	
RX	現在の向き（度）	
DX	進む長さ（ドット）	
C	線の色	
PS	＝0	線を引かずに移動する
	＋	線を引く

また、POSITIONで画面の座標は次のようになっています。

角度の指定は次のようにX軸方向が0°で、時計と逆まわりに増えていきます。

行番号140から190を変えて、いくつかプログラムをつくってみました。

＊4・12　タートルの動きをつくる

```
100 REM *** 4.12 ***
110 REM *** ﾒｲﾛ ﾐﾀｲﾀﾞ ﾅ ***
120 SCREEN 2,2:CLS
130 POSITION (128,96),0,1
140 C=1:PS=1
150 X0=0:Y0=0:RX=0:DX=10
160 FOR I=1 TO 16
170 GOSUB 1020
180 RX=RX+90:DX=DX+10
190 NEXT I
200 GOTO 200
1000 REM ***FORWARD***
1010 REM ***X0,Y0,RX,DX,C,PS***
1020 RX=RX MOD 360
1030 PP=PI/180
1040 X1=X0:Y1=Y0
1050 X0=X0+COS(RX*PP)*DX
1060 Y0=Y0+SIN(RX*PP)*DX
1070 IF PS=0 THEN RETURN
1080 LINE (X1,Y1)-(X0,Y0),C
1090 RETURN
```

プログラムを実行すると、次ページのような絵を描きます。

次はこんなプログラムです。いったいどんな形が現われるでしょうか？

＊4・13　さあ、実行してみよう！

```
100 REM *** 4.13 ***
110 REM *** ナニ ガ カケルカナ? ***
120 SCREEN 2,2:CLS
130 POSITION (128,96),0,1
140 C=1:PS=1
150 X0=0:Y0=60:RX=210:DX=60
160 FOR I=1 TO 6
170 GOSUB 1020
180 RX=RX+60
190 NEXT I
200 GOTO 200
1000 REM ***FORWARD***
1010 REM ***X0,Y0,RX,DX,C,PS***
1020 RX=RX MOD 360
1030 PP=PI/180
1040 X1=X0:Y1=Y0
1050 X0=X0+COS(RX*PP)*DX
1060 Y0=Y0+SIN(RX*PP)*DX
1070 IF PS=0 THEN RETURN
1080 LINE (X1,Y1)-(X0,Y0),C
1090 RETURN
```

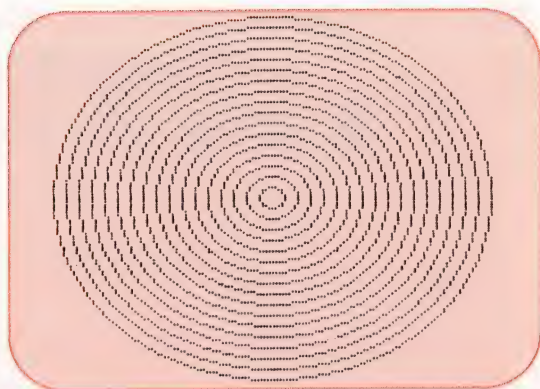
12 円を描く

さて、これまでは直線を使った図形が多かったので、このへんで円を描くプログラムを紹介しましょう。

ここではPSET(ピーセット)を使ってちょっとずつ点を打ち、円を描いていきます。PSETというのは、画面上に点を描いていく働きをするものです。

＊4・14　まん丸の円を描いてみよう

```
100 *** 4.14 ***
110 *** メ ガ マワッチャウ! ***
120 SCREEN 2,2:CLS
130 X=128:Y=96:CO=1
140 FOR R=4 TO 90 STEP 4
150 GOSUB 1000
160 NEXT R
170 GOTO 170
1000 REM ***CIRCLE***
1010 CX=R:CY=0:F=INT(-R/2)
1020 IF CY>CX THEN RETURN
1030 PSET (X+CX,Y+CY),CO
1040 PSET (X-CX,Y+CY),CO
1050 PSET (X+CX,Y-CY),CO
1060 PSET (X-CX,Y-CY),CO
1070 PSET (X+CY,Y+CX),CO
1080 PSET (X-CY,Y+CX),CO
1090 PSET (X+CY,Y-CX),CO
1100 PSET (X-CY,Y-CX),CO
1110 CY=CY+1:F=F+CY
1120 IF F>0 THEN F=F-CX:CX=CX-1
1130 GOTO 1020
```

では、実行――。

プログラムを実行すると、小さな円から大きな円まで、何重にも円を描いていきます。

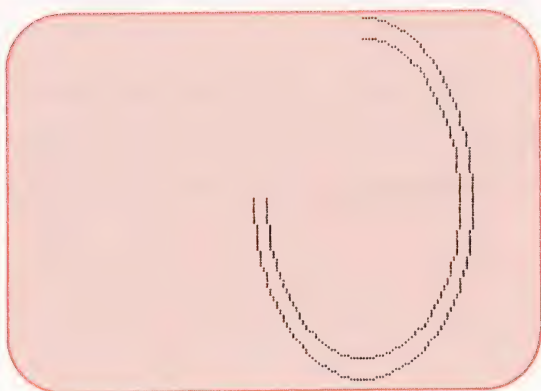
これらの1つずつの円を描いているのは、行番号1000以降のプログラムです。

また、円の大きさや色は、XやY、COで指定していますので、自分で自由に設定してください。

さて、ただの円だけじゃ満足できないという人もいるでしょう。卵のような形やお椀のようなものもつくりたい、と思ったら、次のプログラムを入力してください。

これは、だ円を描くプログラムです。このうち、だ円を描いているのは、行番号1000以降です。これも、変数X、Y、CO、RA、ST、ENを自分で設定し、行番号1040以降を実行すれば、だ円の大きさを自由に変えることができます。

＊4・15　だ円を描こう

```
100 REM *** 4.15 ***
110 REM *** ﾀﾏｺﾞ ﾀﾞｯﾃ ｶｹﾙﾖ ***
120 SCREEN 2,2:CLS
130 X=128:Y=96:CO=1:RA=2
135 ST=.75:EN=.5
140 FOR R=30 TO 40 STEP 4
150 GOSUB 1000
160 NEXT R
170 GOTO 170
1000 REM *********CIRCLE
1010 REM *********X,R,CO,RA,ST,EN
1040 IF ST>EN THEN ST=ST-1
1050 C0=ST*2*PI
1055 PSET(X+COS(C0)*R+.5,Y+SIN(C0)*R*R
A+.5)
1060 FOR C1=C0 TO EN*2*PI STEP PI/2/R
1070 LINE -(X+COS(C1)*R+.5,Y+SIN(C1)*R
*RA+.5),CO
1080 NEXT C1
1090 RETURN
```

この角度のヒミツは？

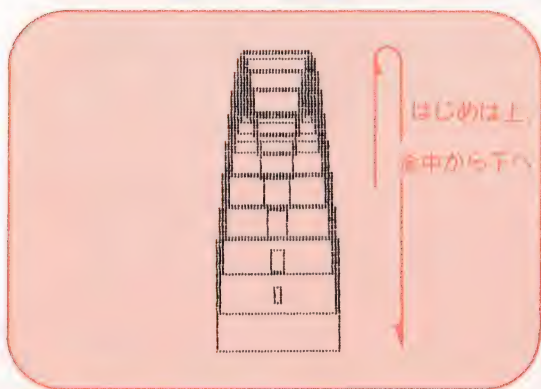
13 四角形を使って グラフィックをつくる

最後に、ちょっと本格的なコンピュータ・グラフィックに入門。

＊4・16　ちょっとややこしいけどガンバレ！

```
5 REM *** 4.16 ***
8 REM *** ゴジュウ ノ トウ モ カンタンダ ***
10 SCREEN 2,2:CLS
20 COLOR 1,15,(0,0)-(255,191),5
30 REM
40 SCREEN 2,2:CLS
50 COLOR 1,15,(0,0)-(255,191),5
60 X=80:Y=90:XX=120:YY=130:E=16:F=16
70 FOR T=180 TO 0 STEP -5
80 S=INT(SIN(RAD(T))*80)
90 LINE(X+E,Y+F-S)-(XX-E,YY-F-S),5,B
100 FOR W=0 TO 1 :NEXT W
110 BLINE(X+E,Y+F-S)-(XX-E,YY-F-S),1,B
120 E=E+1:F=F+1
130 NEXT T
140 REM
150 SCREEN 2,2:CLS
160 X=80:Y=140:XX=120:YY=180:E=1:F=1
170 COLOR 1,15,(0,0)-(255,191),5
180 REM
190 FOR T=0 TO 180 STEP 10
200 S=INT(SIN(RAD(T))*80)
210 LINE(X+E,Y-F-S)-(XX-E,YY+F-S),5,B
230 E=E+1:F=F-1
240 NEXT T
250 GOTO 40
```

このプログラムを実行すると四角形を上方向へ次々に描いていきます。四角形はだんだん小さくなっていきます。

途中までいくと、今度は下方向へ次々に描きはじめます。

どうですか？　ちょっとしたもんだナと思いませんか？

5

もっともっとアニメらしく

自分のつくったオリジナルアニメが画面の中を動きまわるゾ！

スイッチオン！ する前に

画面上を、単純なキャラクタが動きまわるだけでは、ちょっと物足りませんね。もっと複雑なキャラクタをつくって、しかも画面の中を自由に動かしたいものですね。ここでは、そういうこまかい動きをさせてみることにしましょう。ぐっとアニメーションらしくなって、楽しさもグ～ンと増すはずですよ。

1 組み合わせたキャラクタを動かす

キャラクタを組み合わせたものを動かすだけでも、楽しいアニメがつくれます。

まずは、簡単にキャラクタを組み合わせて、１つの図形をつくり、それを動かすプログラムを見てみましょう。

＊５・１　キャラクタを組み合わせて１つの図形をつくる

```
100 REM *** 5.1 ***
110 REM *** ｸﾙﾏ ｶﾞ ｳｺﾞｸﾖ ***
120 CLS
130 X=36:Y=1
140 GOTO 200
150 CURSOR X,Y:PRINT "  ";
160 X=X-2
170 Y=Y+1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT "H";
210 FOR T=1 TO 40:NEXT T
220 GOTO 150
230 GOTO 120
```

上のプログラムは、グラフィックキーの□と□を組み合わせて１つのキャラクタ“H”をつくり、書いたり消したりすることで、それを動かしています。

このキャラクタは、キーをヨコに２つ並べた形なので、前に表示したキャラクタを消すためには、行番号150で、キャラクタの長さ分、つまりスペースを２つ入れています。

次は、グラフィックキーをタテに組み合わせたキャラクタが動くプログラムです。

＊5・2　タテに組み合わせる

```
100 REM *** 5.2 ***
110 REM *** ﾄﾗﾝﾌﾟ ﾉ ﾀﾞｲﾔ ｶﾞ ｳｺﾞｸﾖ ***
120 CLS
130 X=36:Y=1
140 GOTO 200
150 CURSOR X,Y:PRINT "  ";
155 CURSOR X,Y+1:PRINT "  ";
160 X=X-2
170 Y=Y+1
180 IF X<0 OR X>37 THEN 230
190 IF Y<0 OR Y>23 THEN 230
200 CURSOR X,Y:PRINT "◢◣";
205 CURSOR X,Y+1:PRINT "◥◤";
210 FOR T=1 TO 40:NEXT T
220 GOTO 150
230 GOTO 120
```

プログラムを実行すると、"◆"が動きますね。この場合も、前に表示したキャラクタを消していくスペースが必要です。"◆"は、◢、◣、◥、◤の4つのキーを組み合わせてありますので、行番号150と155を使って、

タテの分のスペースも入れています。

形の変わらない図形が動くだけでは、あまり面白くありませんね。キャラクタが変化しながら動いたり、違うものに変身しながら動いていったら、これはもう、立派なアニメーションです。

馬が走る、というアニメーションを考えてみましょう。

＊５・３　馬がパッカパッカと走る

```
100 REM *** 5.3 ***
110 REM *** ｳﾏ ｶﾞ ﾊﾟｯｶﾊﾟｯｶ ﾊｼﾙﾖ ***
120 CLS
130 X=6:Y=10:F=1
140 L1$="    ／ ＼":L2$="    ＼ ／"
150 CURSOR X-2,Y:PRINT" ██"
160 CURSOR X-2,Y+1:PRINT" █"
170 CURSOR X-5,Y+2:PRINT" ／███"
180 CURSOR X-6,Y+3
190 IF F=1 THEN PRINT L1$
200 IF F=-1 THEN PRINT L2$
210 FOR T=1 TO 20:NEXT T
220 X=X+1
230 F=-F
240 IF X>37 THEN 120
250 GOTO 150
```

上のプログラムを実行すると、馬が足を動かしながら走ります。馬のキャラクタは、１章でつくりましたね。このプログラムでは、行番号140に、向きの違う足の形を２種類入れてあります。それを、行番号190と200で、交互に表示するようにして、走っているように見せています。

さて、次のプログラムでは、キャラクタを組み合わせた人間が２人登場します。さあどうなるか、入力してみてください。

＊5・4　人間が２人登場！

```
5 REM *** 5.4 ***
8 REM *** フェンシング゛ ノ シアイ ダ゛ヨ ***
10 CLS:COLOR 5,15
15 X=15:Y=10:XX=25:E=1:F=1
20 A$="  *  ":B$=" ┗0─ ":C$="  ^  "
25 AA$="  ♨  ":BB$=" ─■┛ ":CC$="   ^
 "
26 COLOR 8,15
30 CURSORX,Y:PRINTA$
40 CURSORX,Y+1:PRINTB$
50 CURSORX,Y+2:PRINTC$
60 E=INT((RND(1)*3)-1)
70 IF X<=5 OR X>=30 THEN X=X-(E*2)
80 X=X+E
90 BEEP
100 COLOR 5,15
130 CURSORXX,Y:PRINTAA$
140 CURSORXX,Y+1:PRINTBB$
150 CURSORXX,Y+2:PRINTCC$
160 F=INT((RND(1)*3)-1)
170 IF XX<=5 OR XX>=30THENXX=XX-(F*2)
180 XX=XX+F
190 GOTO 30
```

ボクシングも
スモウも
できそうダ

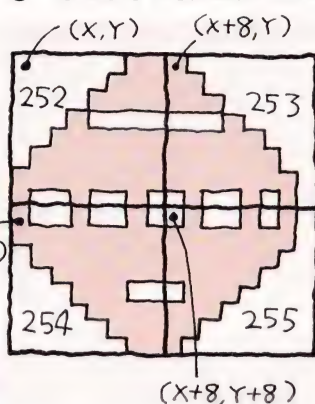
2 UFOをつくってキーで動かす

PATTERN文を使ってキャラクタをつくる、というのは、１章でやってみましたね。ここでも、PATTERN文を使ってキャラクタをつくるのですが、今度は、グラフィック画面でつくるので、8×8ドットすべてを使えます。

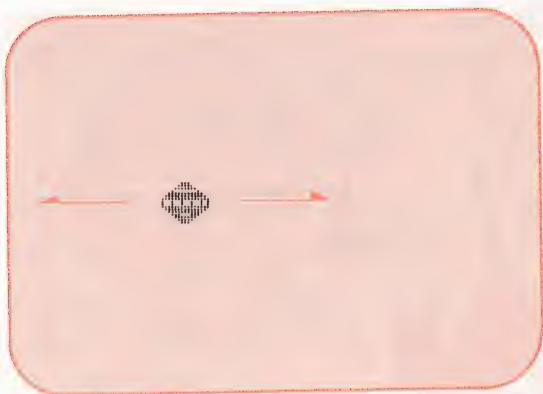
まず、キャラクタをつくって表示するプログラムを考えてみましょう。

＊5・5　お待ちかね、UFOをつくろう！

```
100 REM *** 5.5 ***
110 REM *** UFO ダ! ***
120 SCREEN 2,2:CLS
130 X=100:Y=100
140 PATTERN C#252,"03070F103F7FFF92"
150 PATTERN C#253,"80C0E010F8FCFE4A"
160 PATTERN C#254,"92FF7F3F1C0F0703"
170 PATTERN C#255,"4AFEFCF870E0C080"
180 CURSOR X,Y:PRINT CHR$(252)
190 CURSOR X+8,Y:PRINT CHR$(253)
200 CURSOR X,Y+8:PRINT CHR$(254)
210 CURSOR X+8,Y+8:PRINT CHR$(255)
220 GOTO 220
```

上のプログラムを実行すると、ＵＦＯを、画面のまん中に表示します。ＵＦＯは、４つのキャラクタを組み合わせてつくっています。キャラクタを16進数に直す方法は、１章で使ったテキスト画面のときと同じです（25ページにあります）。

次に、このUFOを動かすプログラムをつくってみましょう。

＊5・6　UFOを動かす

```
100 REM *** 5.6 ***
110 REM *** UFO ヲ ウゴカシテミヨウ ***
120 SCREEN 2,2:CLS
130 X=100:Y=100:V=5
140 PATTERN C#252,"03070F103F7FFF92"
150 PATTERN C#253,"80C0E010F8FCFE4A"
160 PATTERN C#254,"92FF7F3F1C0F0703"
170 PATTERN C#255,"4AFEFCF870E0C080"
180 GOTO 230
190 BLINE (X,Y)-(X+16,Y+16),,BF
200 X=X+DX
210 IF X>239 THEN X=239
220 IF X<0 THEN X=0
230 CURSOR X,Y:PRINT CHR$(252)
240 CURSOR X+8,Y:PRINT CHR$(253)
250 CURSOR X,Y+8:PRINT CHR$(254)
260 CURSOR X+8,Y+8:PRINT CHR$(255)
270 A$=INKEY$:IF A$="" THEN 270
280 IF A$=CHR$(29) THEN DX=-V
290 IF A$=CHR$(28) THEN DX=V
300 GOTO 190
```

UFOが
ヒダリへ、ミギへ

プログラムを実行すると、←と→を使って、UFOを左右に動かせます。この場合も、UFOを描いたり消したりして動かしています。ただし、今までのように、スペースを使って、前に表示したキャラクタを消しているのではなく、BLINE（ビーライン）で消しているのがミソです。BLINEは、行番号190に入っていますね。

行番号130の変数Vは、キーを1回押すごとに、UFOが動くドット数です。このプログラムでは、1回に5ドットずつ動いています。ドット数を大きくするほど、動く速さは速くなります。ですからドット数を小さくすると、当然動きはゆっくりになりますね。変数Vの値を変えて、いろいろためしてみてください。

3 ウサギが上から降りてくる

今度も、グラフィック画面で、キャラクタを組み合わせて、1つのものをつくりました。次のプログラムを入力してみてください。

＊5・7　動物をつくって動かす

```
5 REM *** 5.7 ***
8 REM *** ｳｻｷﾞ ｶﾞ ｵﾘﾃｸﾙｿﾞ ***
10 CLS:X=20:Y=3
20 PATTERNC#240,"081818181818181C"
30 PATTERNC#241,"1C1434FC3CF8381C"
40 PATTERNC#242,"40606060606060E0"
50 PATTERNC#243,"E0A0B0FCF07C70E0"
60 A$=CHR$(240):B$=CHR$(241)
70 C$=CHR$(242):D$=CHR$(243)
80 COLOR 8,15
90 CURSORX,Y:PRINTA$
100 CURSORX,Y+1:PRINTB$
110 CURSORX+1,Y:PRINTC$
120 CURSORX+1,Y+1:PRINTD$
130 FOR W=0 TO 80:NEXT W
140 COLOR 14,15
150 CURSORX,Y:PRINT" "
160 CURSORX,Y+1:PRINT" "
170 CURSORX+1,Y:PRINT" "
180 CURSORX+1,Y+1:PRINT" "
190 Y=Y+1:IF Y>=22THEN Y=0
200 GOTO 80
```

ウサギが
上から下へ
スウウウウ………

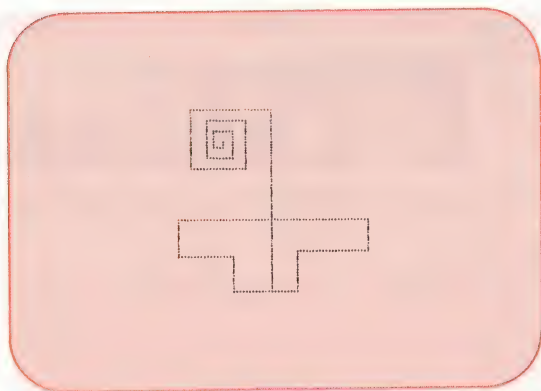
4 点を動かすとお絵描きができる

今までやってきたように、グラフィック画面を使って絵を描くと、テキスト画面よりも、こまかい絵が描けます。これと同じように、グラフィック画面でキャラクタを動かすと、テキスト画面よりもきめこまかな動きをします。

次のプログラムは、点を動かして、よりこまかな絵を描くことができます。そういえば、3章50ページに、同じようなプログラムがありましたね。

＊5・8　お絵描きプログラム

```
100 REM *** 5.8 ***
110 REM *** フクザツナ エ モ カケルヨ ***
120 SCREEN 2,2:CLS
130 X=125:Y=96
140 PSET (X,Y)
150 A$=INKEY$:IF A$="" THEN 150
160 IF A$=CHR$(30) THEN Y=Y-1
170 IF A$=CHR$(31) THEN Y=Y+1
180 IF A$=CHR$(29) THEN X=X-1
190 IF A$=CHR$(28) THEN X=X+1
200 IF X<0 OR X>255 OR Y<0 OR Y>191 TH
EN 130
210 PSET (X,Y)
220 GOTO 150
230 NEXT X
240 GOTO 240
```

チューリップ、
カーネーション……
何でもできるヨ

↑↓←→を押して、点を動かしてみましょう。とてもこまかい動きをしますね。これは、グラフィック画面を使って、点が1ドットずつ動くようにしてあるからです。グラフィック画面だと、このようにして、ちょっと複雑なこまかい絵が描けます。ためしてみてください。

5 線を動かすと ビーム発射になる

ゲームセンターにあるゲームでおなじみ、ビーム光線をつくってみましょう。

まず、画面にナナメの直線を引きます。次のプログラムを入力してください。

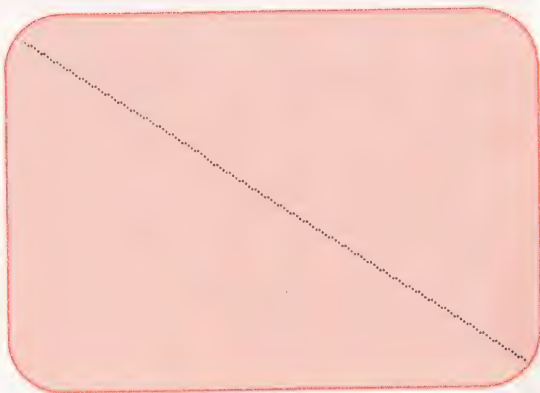
＊5・9　ナナメに直線を引く

```
100 REM *** 5.9 ***
110 REM *** ビームコウセン ダヨ ***
120 SCREEN 2,2:CLS
130 X1=0:Y1=0:X2=255:Y2=191
140 C=1
150 GOSUB 1020
160 GOTO 160
1000 REM ***LINE***
1010 REM ***X1,Y1,X2,Y2,C
1020 DX=X2-X1:DY=Y2-Y1
1030 ST=ABS(DX)+1:IF ABS(DX)<ABS(DY) T
HEN ST=ABS(DY)+1
1040 SX=DX/ST:SY=DY/ST
1050 X0=X1+0.5:Y0=Y1+0.5
1060 FOR UU=1 TO ST+1
1070 PSET(X0,Y0),C
1080 X0=X0+SX:Y0=Y0+SY
1090 NEXT UU
1100 RETURN
```

プログラムを実行すると、画面にナナメの線を引きます。

このプログラムの行番号1080と1090の間に、行番号1085を挿入してみてください。

```
1085 A$=INKEY$: IF A$= "" THEN 1085
```

行番号1085を挿入（そうにゅう）したプログラムを実行したら、どのキーでもかまいませんので、好きなキーを押してみてください。ナナメの短い直線が描けましたね。キーを押し続けるか、または何度も押すと、さっきと同じナナメの直線が引けます。このようにして、少しずつ線を描いていくこともできます。

グラフィック画面では、キャラクタを動かすほかに、線を動かすこともできます。まずは、次のプログラムを入力してみてください。

＊5・10　ビーム光線が左から右へ

```
100 REM *** 5.10 ***
110 REM *** ﾋﾞｰﾑｺｳｾﾝ ﾊｯｼｬ! ***
120 SCREEN 2,2:CLS
130 X=0:Y=100
140 PSET (X,Y)
150 X=X+10
160 IF X>255 THEN 120
170 LINE -(X,Y)
180 GOTO 150
```

上のプログラムを実行すると、左から右へ線が走っていきます。なんとなくビーム光線を発射（はっしゃ）しているように見えますね。

キーを押すと、ビーム光線を発射する、というふうにできたら、ゲームに使えそうです。それには、INKEY$（インキーダラー）を使います。

次のプログラムでは、ビーム光線をタテに発射するようにし、さらに、キーを押すとビームを発射するようにしてみました。

＊5・11　ビーム光線発射!

```
100 REM *** 5.11 ***
110 REM *** ビームコウセン ヲ タテニ ハッシャ ***
120 SCREEN 2,2:CLS
130 X=128:Y0=191:Y1=10:S=-10
140 A$=INKEY$:IF A$="" THEN 140
150 PSET (X,Y0)
160 FOR Y=Y0 TO Y1 STEP S
170 LINE -(X,Y)
180 FOR T=1 TO 5:NEXT T
190 NEXT Y
200 LINE -(X,Y1)
210 BLINE (X,Y0)-(X,Y1)
220 GOTO 140
```

上のプログラムを実行して、キーをどれでも１つ押すと、下から上へビーム光線を発射します。行番号140で、INKEY$を使っていますね。

さて、ビーム光線を発射するプログラムを応用して、短い直線が飛んでいく(ビームが飛んでいく)プログラムをつくってみました。次のプログラムがそれです。

＊5・12　ビームが飛んでいく

```
100 REM *** 5.12 ***
110 REM *** ビーム ガ ソラヲ トンデユクヨ ***
120 SCREEN 2,2:CLS
130 X=10:Y=100
140 LINE (0,Y)-(X,Y)
150 X=X+10
160 IF X>255 THEN 120
170 LINE (X-10,Y)-(X,Y)
180 BLINE (X-20,Y)-(X-10,Y)
190 FOR T=1 TO 10: NEXT T
200 GOTO 150
```

実行すると、短い直線（ビーム）が左から右へ飛んでいきます。これは、前に表示した線を、行番号180でBLINEを使って消しているからです。

次は、キーを押すと、ビームがタテに飛んでいくプログラムです。

＊5・13　ビームがタテに飛んでいく

```
100 REM *** 5.13 ***
110 REM *** ビ-ム ガ タテニ トンデユクヨ ***
120 SCREEN 2,2:CLS
130 X=128:Y0=191:Y1=10:S=-5:L=10
140 A$=INKEY$:IF A$="" THEN 140
150 LINE (X,Y0)-(X,Y0-L)
160 FOR Y=Y0+S TO Y1 STEP S
170 LINE (X,Y)-(X,Y-L)
180 BLINE (X,Y-S)-(X,Y-1)
190 NEXT Y
200 BLINE (X,Y-S)-(X,Y1-L)
210 GOTO 140
```

プログラムを実行したら、キーをどれでも1つ押してみましょう。ビームが下から上へ飛んでいきます。キーを押すとビームが飛んでいくのは、前にも説明したとおり、INKEY$を使っているからです。このプログラムでは、行番号140でINKEY$を使っています。

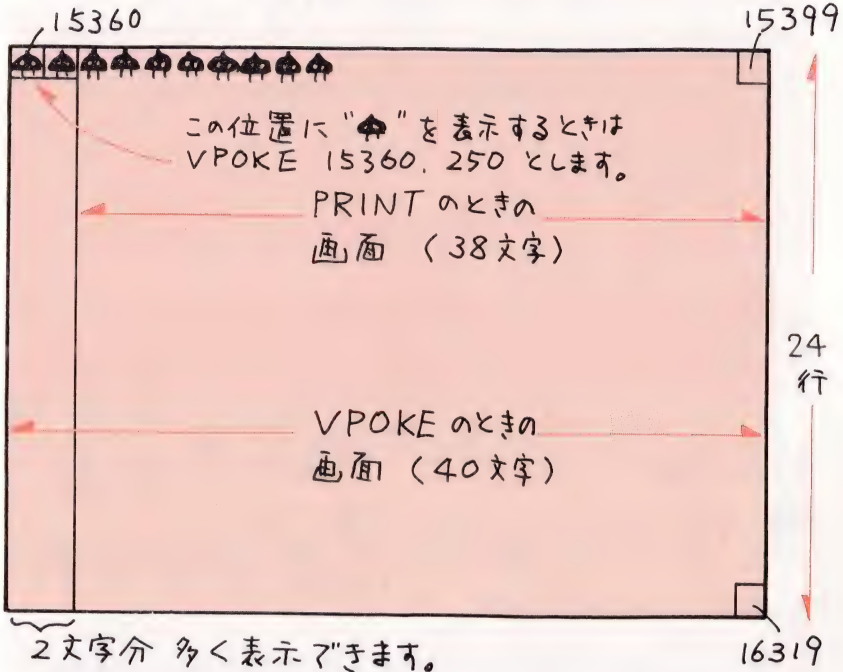
6 VPOKEを使うと画面いっぱい使える

今までは、キャラクタを画面に表示する方法として、PRINTを使ってきました。ここでは、VPOKE(ブイポーク)を使って、キャラクタを表示してみましょう。

＊5・14　キャラクタを画面いっぱいに表示する

```
100 REM *** 5.14 ***
110 REM *** ｴｲﾘｱﾝ ﾄｳｼﾞｮｳ! ***
120 CLS
130 V=15360
140 FOR I=V TO V+10
150 VPOKE I,250
160 NEXT I
170 GOTO 170
```

プログラムを実行すると、画面の左端から"♠"を10個表示します。VPOKEは、PRINTと同じように、キャラクタを表示する働きをもっています。ただし、VPOKEでは、画面上の1行に40文字表示できます。PRINTを使ったときより、1行につき2文字分多く表示できるわけです。わかりやすく、図にしてみると、右のようになります。

何コでも
出せるよ

PRINTを使ってキャラクタを表示するときは、CURSORでXとYの位置を指定しましたが、VPOKEでは、キャラクタの位置を15360～16319の数で指定します。また、表示したいキャラクタも、キャラクタコードで指定します。表示したいキャラクタと位置を、VPOKEだけで指定できるので、PRINTとCURSORを使うよりも便利ですね。

ところで、キャラクタをタテ、ヨコ、ナナメ自由自在に動かす方法については、2章でPRINTを使ってやってみました。今度は、VPOKEでキャラクタを動かしてみましょう。VPOKEを使う場合も、キャラクタを表示する位置を変えて、描いたり消したりすれば、動かせるはずです。

次のプログラムを入力してみましょう。

＊5・15　キャラクタを描いたり消したりする

```
100 REM *** 5.15 ***
110 REM *** VPOKE デ ウゴカシテミヨウ ***
120 CLS
130 V=15760
140 VPOKE V,32
150 V=V+1
160 IF V>15799 THEN 120
170 VPOKE V,236
180 FOR T=1 TO 30:NEXT T
190 GOTO 140
```

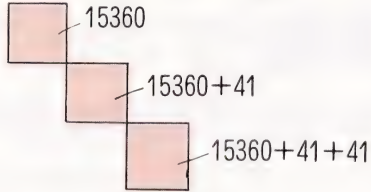
プログラムを実行すると、"●"が画面を左から右へよこぎります。行番号140で指定しているキャラクタコードの32は、スペースです。また、行番号170のキャラクタコード236は、"●"です。

このプログラムでは、最初にキャラクタを表示する位置と最後に表示する位置は、次のようになります。

それでは、VPOKEを使って、キャラクタをナナメに動かすにはどうすればいいのでしょう。まず、次のプログラムを入力してください。

＊5・16　キャラクタをナナメに動かす

```
100 REM *** 5.16 ***
110 REM *** VPOKE デ ナナメニ ウゴカス ***
120 CLS
130 V=15360
140 VPOKE V,32
150 V=V+41
160 IF V>16319 THEN 120
170 VPOKE V,236
180 FOR T=1 TO 50:NEXT T
190 GOTO 140
```

プログラムを実行すると、"●"が画面をナナメに動きます。VPOKEでの1行は40文字ですので、このプログラムでは、行番号150でV＝V＋41とすることによって、キャラクタをナナメ右下に動かしています。

ところで、タテ、ヨコ、ナナメの単純な動きだけではなく、いろいろと角度を変えて、キャラクタを動かすにはどうすればよいのでしょうか。

PRINTでは、変数XとYを使いましたね。VPOKEでも、やはり変数XとYを使います。

次のプログラムを入力してください。

＊5・17　いろいろと角度を変える

```
100 REM *** 5.17 ***
110 REM *** ｴｲﾘｱﾝ ｶﾞ ﾔｯﾃｷﾀ! ***
120 CLS
130 V=15360
140 X=11:Y=22
150 GOTO 210
160 VPOKE V+X+Y*40,32
170 X=X+3
180 Y=Y-1
190 IF X<0 OR X>37 THEN 120
200 IF Y<0 OR Y>23 THEN 120
210 VPOKE V+X+Y*40,250
220 FOR T=1 TO 40:NEXT T
230 GOTO 160
```

プログラムを実行すると、"♠"がナナメ右上に動いていきます。

キャラクタのX座標をX、Y座標をYとすると、VPOKEを使ったときのキャラクタの位置は、次の式で求めることができます。

15360＋X＋Y×40

前のページのプログラムでは、行番号160でこの式を使っていますね。あとは、PRINTのときと同じように、行番号170と180で、数字を変えれば、角度を変えてキャラクタを動かせます。

6

アイデアとテクニックで勝負！

ゲームをおもしろくするもしないも、キミのアイデア次第！

スイッチオン！　する前に

“ブロックくずし”はどうするんだろう、“インベーダーゲーム”の砲台は、ミサイルは、バクハツの状態は……ゲームをつくるさい疑問に思う部分はたくさんありますね。そこで、この章では、ゲームづくりのための、ゲームをおもしろくするとっておきのテクニックとアイデアをご披露(ひろう)しましょう。

1 背景はこうやって動かす

「お父さん、やっぱり画面が動いたほうがゲームらしくなると思うナ」

「そのとおり、これもプログラムを組めばカンタンなんだ」

…………

画面の動きには、キャラクタそのものを動かすのと、背景を動かすことでキャラクタが動いているように見せることの２通りがあります。限られた画面の大きさの中で、上から下に、右から左に動くよりも、画面そのものが動いていくほうが距離感が楽しめるというゲームもあります。そこで、画面の背景となる部分を動かすための画面スクロールの方法を紹介しましょう。

まずは、次のプログラムを実行してみます。

＊6・1　画面スクロールのプログラム

```
100 REM *** 6.1 ***
110 REM *** ﾎｼ ｶﾞ ﾅｶﾞﾚﾃｲｸﾖ ***
120 CLS
130 CURSOR 10,23:PRINT " ※ ※ ※ ※"
140 FOR T=1 TO 40:NEXT T
150 CURSOR 10,23:PRINT "※ ※ ※ ※ "
160 FOR T=1 TO 10:NEXT T
170 GOTO 130
```

〝※〟が下から上に流れていくように見えますね。これが、画面スクロールです。

実際にスクロールが起きているのは、行番号130と150です。画面の一番下の行に、〝※〟を表示しては改行するといったPRINT（プリント）の働きで、どんどん上に流れていく、これがスクロールです。

次は、スクロールしていく画面の中で、キャラクタが停止している状態、つまり、キャラクタは動かずに背景が動いていくプログラムをみましょう。

＊６・２　キャラクタは動かずに背景だけが動く

```
100 REM *** 6.2 ***
110 REM *** ﾙｰﾑﾗﾝﾅｰ ﾐﾀｲﾀﾞ ***
120 CLS
130 CURSOR 10,23:PRINT " * * * *"
140 CURSOR 20,17:PRINT"工"
150 FOR T=1 TO 10:NEXT T
160 CURSOR 20,17:PRINT" "
170 CURSOR 10,23:PRINT "* * * * "
180 CURSOR 20,17:PRINT"工"
190 FOR T=1 TO 10:NEXT T
200 CURSOR 20,17:PRINT" "
210 GOTO 130
```

“工”は動かずに、背景の“＊”だけが流れていきます。まるで、ルームランナーのようではありませんか。

2 カーレースの道路はこうつくる

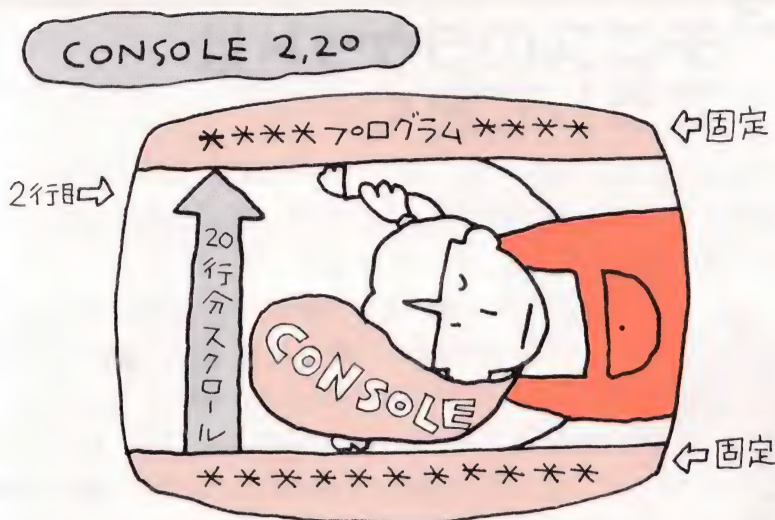
ゲームセンターにあるグランプリレースのゲーム。あのゲームは、画面スクロールを最も効果的に使っているゲームです。あの距離感、あのスピード感、ぜひ手づくりのゲームで味わいたいものです。

こんどは、部分スクロールを利用して、画面の一部分に道路をつくり、そこだけにスクロールを起こします。

さっそく、次のプログラムを実行してみましょう。

＊6・3　部分スクロールを利用する

```
100 REM *** 6.3 ***
110 REM *** ｶｰﾚｰｽ ﾖｳ ﾉ ﾄﾞｳﾛ ﾀﾞﾖ ***
120 CLS
130 CONSOLE 0,24
140 CURSOR 6,1:PRINT"*********ﾌﾟﾛｸﾞﾗﾑ 6
.3*********"
150 CURSOR6,22:PRINT"******************
************"
160 CONSOLE 2,20
170 CURSOR 10,21:PRINT " *  *  *  *"
180 CURSOR 19,17:PRINT"  "
190 FOR T=1 TO 10:NEXT T
200 CURSOR 19,17:PRINT"  "
230 FOR T=1 TO 10:NEXT T
250 GOTO 160
```

画面の上と下はそのままで、まん中だけがスクロールしています。これは、行番号160で、CONSOLE（コンソール）2、20として2行目から20行分をスクロールの範囲に決めているからです。

このように、CONSOLE を使うと、範囲を決めた部分スクロールが実現します。スクロールするところは道路に、しないところは得点を表示したりするために、というように使い分けが楽しめます。

また、スクロールの範囲を狭くすると、スロットマシンなんかもつくれそうですね。ゲームはアイデアが勝負です。自分なりに工夫して考えてみてください。

ところで、CONSOLE を１度実行すると、プログラムを止めても部分スクロールはそのままです。元に戻すには、CONSOLE 0、24と入力して、全部の画面がスクロールするように直しておきましょう。

3 テニスのラケットはこうして動く

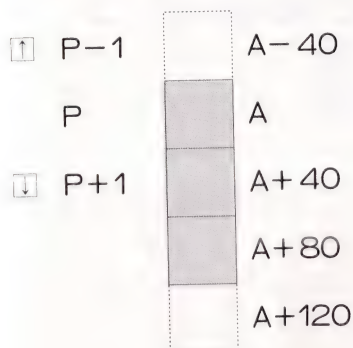
テニスゲームやブロックくずしのラケット、あれはどうやってつくり、どんなふうに動かしているのでしょう。これも、思ったよりカンタンにできてしまいます。

ラケットは、キャラクタの“■”を３つ重ねてつくります。VPOKEを使って、キャラクタコードの229（“■”）をポンポンと画面上に重ねるだけです。そして、これまでに何度もやってきた方法で、カーソルキーを使って上下に、あるいは左右に動かす、これでいいのです。ちょっと気をつけるところは、ラケットが画面からはみ出さないように、画面の両端を判断させることです。

それでは、↑↓で上下にラケットを動かすプログラムを見てみましょう。

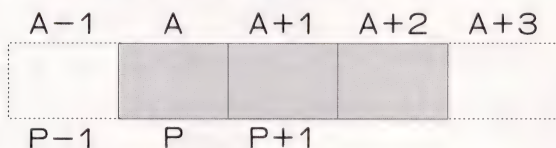
＊６・４　ラケットを上下に動かす

```
100 REM *** 6.4 ***
110 REM *** ﾗｹｯﾄ ｶﾞ ﾀﾃﾆ ｳｺﾞｸﾖ ***
120 CLS
130 P=10
140 GOTO 200
150 A$=INKEY$
160 IF A$=CHR$(30) THEN P=P-1
170 IF A$=CHR$(31) THEN P=P+1
180 IF P<0 THEN P=0
190 IF P>21 THEN P=21
200 A=15397+P*40
210 VPOKE A,229:VPOKE A+40,229:VPOKE A
+80,229
220 VPOKE A-40,32:VPOKE A+120,32
230 GOTO 150
```

プログラムを実行すると、画面の右端にラケットが現われます。↑を押すと上に、↓を押すと下に、ラケットが動きます。このラケットの動き（行番号210と220）を図で示すと、次のようになっています。

これと同じように、←→で左右にラケットを動かすプログラムもつくってみました。

＊6・5　ラケットを左右に動かす

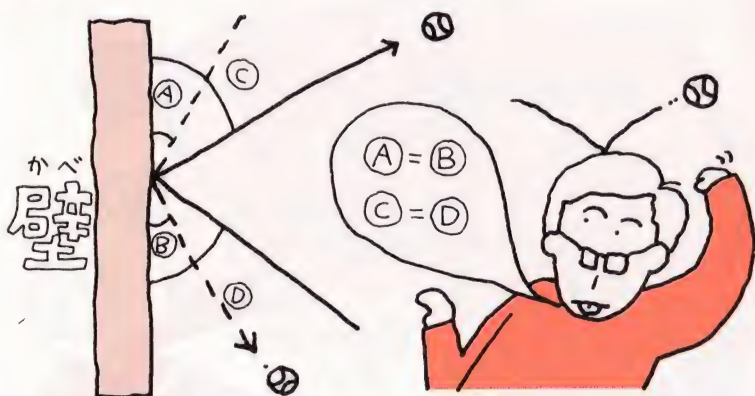
```
100 REM *** 6.5 ***
110 REM *** ﾗｹｯﾄ ｶﾞ ﾖｺﾆ ｳｺﾞｸﾖ ***
120 CLS
130 P=20
140 GOTO 200
150 A$=INKEY$
160 IF A$=CHR$(29) THEN P=P-1
170 IF A$=CHR$(28) THEN P=P+1
180 IF P<0 THEN P=0
190 IF P>35 THEN P=35
200 A=15482+P
210 VPOKE A,229:VPOKE A+1,229:VPOKE A+
2,229
220 VPOKE A-1,32:VPOKE A+3,32
230 GOTO 150
```

実行すると、画面の下方にラケットが現われます。このラケットは、← で左に、→で右に、動かすことができます。ラケットの左右の動き（行番号210と220）を図にすると、こうなります。

画面は下のようになります。

4 ボールのはね返りはこんなふうに

ラケットの次は、ボールの動きです。「こんどはむずかしいゾ」と思っている人も多いのではないでしょうか。壁やラケットに当たったボールは、ただまっすぐ返ってくるのではなく、はね返らねばなりません。ここでは、はね返りに的を絞って説明してみます。

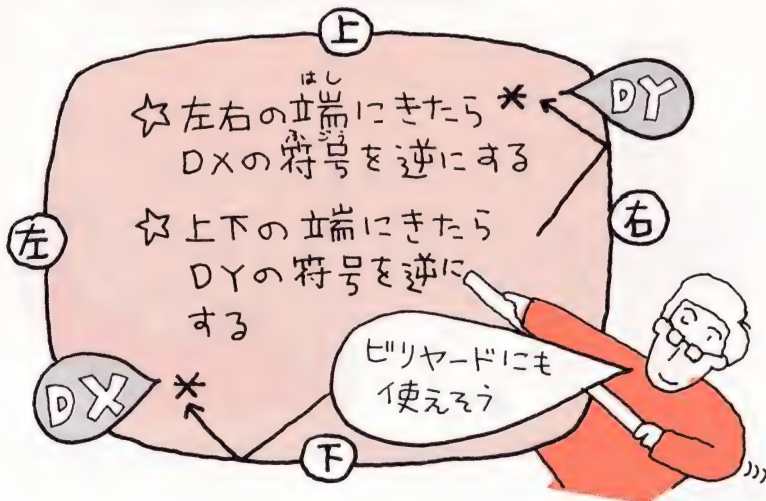
さて、壁に向かってナナメにボールをぶつけると、逆の方向に、やはりナナメにはね返ります。まっすぐぶつけると、まっすぐにはね返ります。ここでちょっと自分が壁になったつもりで考えてみると、上のほうからナナメに飛んできたボールは、下のほうへナナメにはね返り、下のほうから飛んできたならば、上のほうにはね返ることがわかります。

つまり、飛んできた方向とは逆の方向へ、しかも同じ角度ではね返っていくことになります。これを、ちょっと気どって「反射の法則」と呼んでいます。

この反射の法則をプログラムにすると、意外に短くて簡単、次ページのようになります。

＊6・6　はね返りの法則はこのプログラムで

```
100 REM *** 6.6 ***
110 REM *** ﾎﾞｰﾙ ｶﾞ ﾊﾈｶｴﾙﾖ ***
120 CLS
130 X=19:Y=12:DX=1:DY=1
140 GOTO 170
150 CURSOR X,Y:PRINT" ";
160 X=X+DX:Y=Y+DY
170 CURSOR X,Y:PRINT"*";
180 IF X<1 OR  X>36 THEN DX=-DX
190 IF Y<1 OR  Y>22 THEN DY=-DY
200 GOTO 150
```

実行すると、〝＊〟が画面の端にぶつかっては、はね返ります。

プログラムのポイントは、行番号180と190で、DXとDYの符号(ふごう)をひっくり返す（＋から－へ、－から＋へ）ことにあります。画面の上下の端、または左右の端にボールが届いたら、符号を逆にする、これだけで、反射の法則が成り立つのです。

左のプログラムは、CURSORとPRINTの組み合わせで、ボールの動く部分をつくっていますが、もちろんVPOKEを使ってもつくれます。むしろVPOKEのほうが、この場合は使いやすいかもしれません。

VPOKEでは、こんなプログラムになります。

＊6・7　VPOKEを使うと……

```
100 REM *** 6.7 ***
110 REM *** ﾔｯﾊﾟﾘ ﾎﾞｰﾙ ｶﾞ ﾊﾈｶｴﾙﾖ ***
120 CLS
130 X=19:Y=12:DX=2:DY=1
140 GOTO 170
150 VPOKE 15362+X+Y*40,160
160 X=X+DX:Y=Y+DY
170 VPOKE 15362+X+Y*40,236
180 IF X<1 OR  X>36 THEN DX=-DX
190 IF Y<1 OR  Y>22 THEN DY=-DY
200 GOTO 150
```

これを実行すると、次のような画面になります。

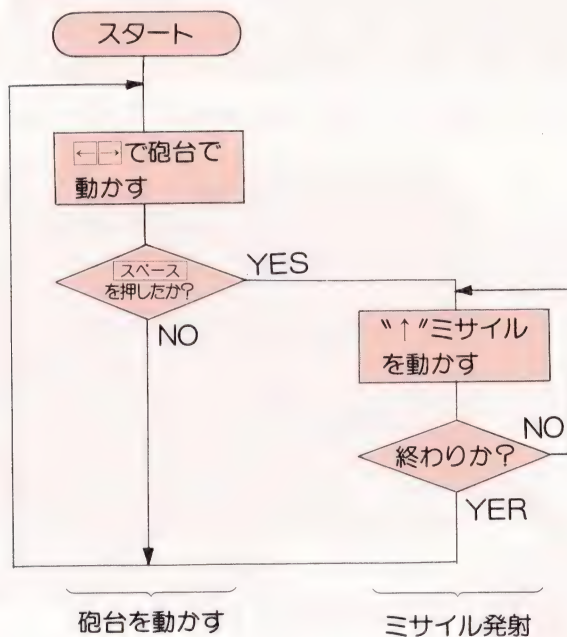
フシギ、フシギ……

5 砲台はこうしてつくる

インベーダーゲーム、ギャラクシアン、どちらも砲台が登場する代表的なゲームです。これらのゲームで夢中で遊んだ覚えのあるお父さんや子供も多いはずです。

さあ、こんどは、この砲台をつくり、弾を発射するところをつくってみましょう。

プログラムの中身は、これまでやってきたことの組み合わせです。とくにあれこれ言うこともありません。キャラクタで砲台をつくり、←→で動かす、そして スペース を押したときだけ、弾の"↑"を発射する、発射

●フローチャートで示すと

した弾は、画面の上まで動いていく。……と、これだけのことです。

そうしてできたプログラムが次のものです。

＊6・8　砲台をつくり、弾を発射する！

```
100 REM *** 6.8 ***
110 REM *** ホウダイ カラ ミサイル ハッシャ! ***
120 CLS
130 X=20
140 GOTO 210
150 A$=INKEY$:IF A$="" THEN 150
160 IF A$=" " THEN 240
170 IF A$=CHR$(28) THEN X=X+1
180 IF A$=CHR$(29) THEN X=X-1
190 IF X<2 THEN X=2
200 IF X>34 THEN X=34
210 CURSOR X-1,22:PRINT" ‖ ";
220 CURSOR X-2,23:PRINT" ▲ ";
230 GOTO 150
240 Y=22
250 VPOKE 15362+X+Y*40,32
260 Y=Y-1
270 VPOKE 15362+X+Y*40,142
280 IF Y<0 THEN 210
290 GOTO 250
```

6 モノにぶつかったかどうかを判断させる

テニスのボールが壁にぶつかったかどうか、大砲の弾が敵に命中したかどうかを判断させることも必要です。

これまでは、画面上の位置（X座標とY座標）でそのへんの判断をやってきましたが、ここでは、VPEEK（ブイピーク）という探知機（たんちき）みたいなものを使って、画面上のキャラクタで判断する方法をプログラムにしてみました。

次のプログラムは、“O（オー）”を並べて壁をつくり、下から飛び出た“●”が壁に接触すると、またはじめから“●”が飛び出します。

＊6・9　壁にあたったかどうか判断させよう

```
100 REM *** 6.9 ***
110 REM *** タマ ハ メイチュウ シタカナ? ***
120 CLS
130 CURSOR 0,5:PRINT"OOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOO"
140 X=20:Y=23
150 GOTO 200
160 VPOKE 15362+X+Y*40,32
170 Y=Y-1
180 C=VPEEK(15362+X+Y*40)
190 IF C=79 THEN 140
200 VPOKE 15362+X+Y*40,236
210 FOR T=1 TO 50 :NEXT T
220 GOTO 160
```

壁にぶつかったかどうかの判定は、行番号180と190です。VPEEKで、移動する先にどんなキャラクタがあるのか確認し、もし壁のO（キャラクタコードは79）だったら、行番号140へ戻る、といった判断をしています。

7 反射の法則を使うとこんなこともできる

＊6・10　ボールが壁に当たるとはね返る

```
100 REM *** 6.10 ***
110 REM *** ﾀﾏ ｶﾞ ｶﾍﾞﾆ ｱﾀﾙﾄ ﾊﾈｶｴﾙﾖ ***
120 CLS
130 CURSOR 0,1:PRINT"OOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOO"
140 CURSOR 0,12:PRINT"OOOOOOOOOOOOOOOOO
OO"
150 X=10:Y=15:DX=1:DY=1
160 GOTO 230
170 VPOKE 15362+X+Y*40,165
180 X=X+DX
190 Y=Y+DY
200 C=VPEEK(15362+X+Y*40)
210 IF C=79 THEN X=X-DX:Y=Y-DY:DY=-DY:
GOTO 180
220 VPOKE 15362+X+Y*40,236
230 IF X<1 OR X>36 THEN DX=-DX
240 IF Y<1 OR Y>22 THEN DY=-DY
250 GOTO 170
```

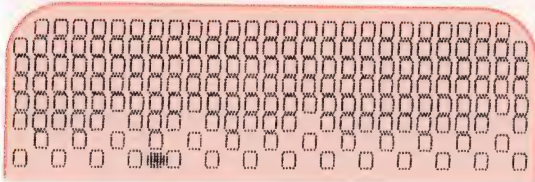
反射の法則と
壁の判断を
使って……

8 ボールがぶつかると消える

ボールがぶつかると消えていくブロック、これも、いままでの応用でつくることができます。

＊6・11　ボールがぶつかると消える

```
100 REM *** 6.11 ***
110 REM *** ﾌﾞﾛｯｸｸｽﾞｼ ｶﾞ ﾃﾞｷﾙﾈ ***
120 CLS
130 CURSOR 0,8
140 FOR I=1 TO 8
150 PRINT"OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO";
160 NEXT I
170 X=10:Y=22:DX=1:DY=-1
180 GOTO 240
190 VPOKE 15362+X+Y*40,32
200 X=X+DX
210 Y=Y+DY
220 C=VPEEK(15362+X+Y*40)
230 VPOKE 15362+X+Y*40,236
240 IF C=79 THEN DY=-DY
250 IF X<1 OR X>36 THEN DX=-DX
260 IF Y<1 OR Y>22 THEN DY=-DY
270 GOTO 190
```

実行すると、ヨコ8列のブロックを表示し、“●”がぶつかったブロックが消えていきます。このプログラムにラケットを追加すれば、もう立派なブロックくずし（?）です。

ブロックくずしだ!!

9 爆発アニメも簡単にできる

弾が命中したら消える、それだけでは「ヤッタ！」という気持ちは味わえません。どうせなら、砕け散る雰囲気を出したいものです。

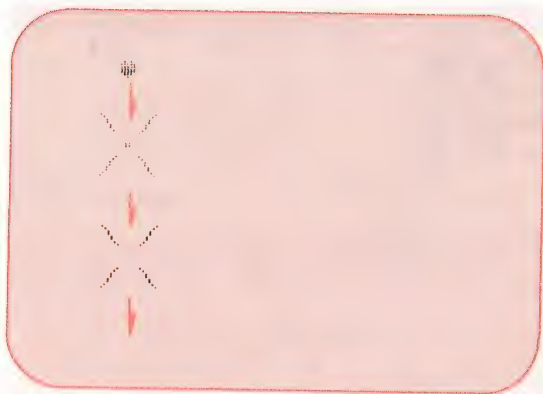
そこで、こんどは、キャラクタが炸裂する状態を、どうやってつくるのかを紹介します。さっそく、プログラムを見てみましょう。

＊6・12　弾が命中するとはじける!!

```
100 REM *** 6.12 ***
110 REM *** ﾀﾏ ｶﾞ ﾒｲﾁｭｳｽﾙﾄ ｷｴﾙﾖ ***
120 CLS
130 X=20:Y=10
140 A=15362+X+Y*40
150 VPOKE A,246
160 A$=INKEY$:IF A$="" THEN 160
170 VPOKE A,165
180 FOR T=1 TO 90:NEXT T
190 VPOKE A-41,148:VPOKE A-39,147:VPOK
E A+39,147:VPOKE A+41,148
200 FOR T=1 TO 90:NEXT T
210 VPOKE A,32
220 FOR T=1 TO 90:NEXT T
230 VPOKE A-41,32:VPOKE A-39,32:VPOKE
A+39,32:VPOKE A+41,32
240 FOR T=1 TO 300:NEXT T
250 GOTO 150
```

プログラムを実行すると、画面に“♥”が現われます。そこで、どれかキーを押すと、ハートは、はじけて消えていきます。

よく見ると、ハートがはじけて消えていくまでに、次のように変化していることがわかります。

このように、表示するキャラクタを次々と変えることで、爆発アニメや炸裂アニメをつくることができます。

ハートがはじけて
消えていくヨ

さて、それでは、この５段階にまたがるキャラクタの変化は、プログラム中のどこにあるのかを見ておきましょう。

まず、〝♥〟の表示は、行番号150です。次の〝・〟は、行番号170で表示し、それに〝　〟、〝／〟、〝／〟、〝　〟を加えて表示したのが、行番号190になります。そして、〝・〟だけを消しているのが行番号210で、最後に全部消してしまうのは行番号230です。

以上の、行番号150、170、190、210、230を適当に手直しして、独自の爆発アニメをつくってみましょう。

でも、まだまだ衝撃が足りないナ、という人は、大きなモノを炸裂させたり、自分でキャラクタパターンをつくって、コナゴナにしたり、いろいろ試してみてください。また、このプログラムを利用して、打ち上げ花火のアニメーションをつくってみるのもいいでしょう。

プログラムづくりのポイントは、キャラクタの変化の間に、適当に時間待ち（FOR～NEXT ですよ）を入れることです。

そうすることで、爆発の状態をタイミングよく表示していくことができます。紹介したプログラムでは、行番号180、200、220において、タイミングをとっています。

10 ミサイルで標的を射ち落とす

前にお話しした砲台とブロックくずしを組み合わせるだけで、標的を射ち落とすプログラムができてしまいました。

＊6・13　標的を射ち落とす

```
100 REM *** 6.13 ***
110 REM *** ミサイル ガ アタルト キエルヨ ***
120 CLS:CURSOR 2,10:PRINT"00000000000
0000000000000000000000"
130 X=20
140 GOTO 200
150 A$=INKEY$:IF A$=" "THEN 230
160 IF A$=CHR$(28) THEN X=X+1
170 IF A$=CHR$(29) THEN X=X-1
180 IF X<2 THEN X=2
190 IF X>34 THEN X=34
200 CURSOR X-1,22:PRINT" ↑ ";
210 CURSOR X-2,23:PRINT" ▲ ";
220 GOTO 150
230 Y=22:C=32
240 VPOKE 15362+X+Y*40,32
250 IF C<>32 THEN 200
260 Y=Y-1:A=15362+X+Y*40
270 C=VPEEK(A):VPOKE A,142
280 IF Y<0 THEN 200
290 GOTO 240
```

←→で砲台を左右に移動し、的を定めて スペース を押すと、ミサイルが発射します。みごと的中すると、その部分の壁は消えてしまいます。

11 エイリアンとヒトが動きまわる

２つ以上のものに別々の動きをさせるプログラムもできるんです。たとえば、エイリアンとヒトが画面上を別々に動きまわっていて、エイリアンはヒトを捜(さが)しまわり、ヒトは逃げまわっているというものです。

このようなプログラムは、次のようになります。

＊６・14　エイリアンとヒトが追っかけっこ！

```
10 REM *** 6.14 ***
20 REM *** ｴｲﾘｱﾝ ｶﾞ ｵｲ ﾋﾄ ｶﾞ ﾆｹﾞﾙ ***
30 CLS
40 X1=1:Y1=1:U1=1:V1=1
50 X2=36:Y2=22:U2=-1:V2=-1
60 GOTO 110
70 VPOKE 15362+X1+Y1*40,165
80 VPOKE 15362+X2+Y2*40,235
90 X1=X1+U1:Y1=Y1+V1
100 X2=X2+U2:Y2=Y2+V2
110 VPOKE 15362+X1+Y1*40,250
120 VPOKE 15362+X2+Y2*40,253
130 IF X1<1 OR X1>36 THEN U1=-U1
140 IF X2<1 OR X2>36 THEN U2=-U2
150 IF Y1<1 OR Y1>22 THEN V1=-V1
160 IF Y2<1 OR Y2>22 THEN V2=-V2
170 GOTO 70
```

実行すると、“♠”と“웃”が現われて、違った足跡を残しながら、別別の動きをします。

プログラムは、単にそれぞれのキャラクタが別々の動きをするように、別々にプログラムにしているだけです。どうってこと、ないですね。

12 シャレたタイトルをつくろう

お店で売っているゲームを買ってきて実行すると、堂々たるタイトルが現われますね。下からずんずんスクロールしていくタイトル、大きく表示されてはゆっくり消えるタイトル、どのゲームも、凝ったタイトルやメッセージを表示して、さらにゲームを盛り上げています。

ここでは、タイトルやメッセージを効果的に表示するプログラムを紹介しましょう。

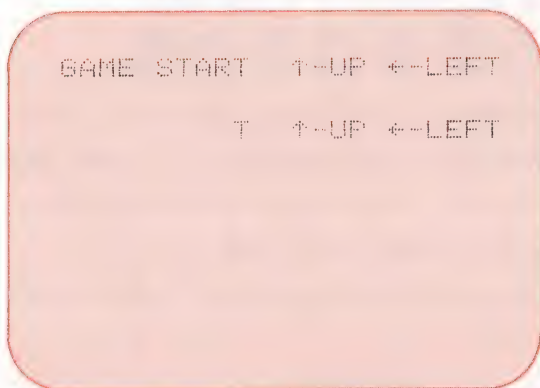
＊6・15　タイトル・メッセージを表示する

```
100 REM *** 6.15 ***
110 REM *** ﾀｲﾄﾙ ﾔ ﾒｯｾｰｼﾞ ｦ ｲﾚﾖｳ ***
120 PRINT CHR$(16)
130 SCREEN 2,2:CLS
140 X=16:Y=80
150 CURSOR X,Y:PRINT "GAME START  ↑-UP
 ←-LEFT"
160 FOR W=0 TO 100:NEXT W
170 FOR X=16 TO 248 STEP 6
180 CURSOR X,Y:PRINT CHR$(8)
190 NEXT X
200 PRINT CHR$(17)
210 X=16
220 CURSOR X,Y:PRINT "12345678 GAME OV
ER"
230 FOR W=0 TO 100:NEXT W
240 FOR X=16 TO 248
250 CURSOR X,Y:PRINT CHR$(8)
260 NEXT X
```

プログラムを実行すると、次ページの画面のように“GAME START ↑-UP ←-LEFT”と表示し、左側から消えます。そして次に、大きな文

字で“12345678 GAME OVER ”と表示し、左側からゆっくり消えていきます。

```
GAME START   ↑-UP ←-LEFT

          T   ↑-UP ←-LEFT
```

ゲームらしく
なってきたネ

これは、コントロールコードのCHR$(16)、CHR$(17)、CHR$(8)を利用してつくっています。

まず、CHR$(16)で、標準サイズの文字を表示させ、それをCHR$(8)で消しています。次に、CHR$(17)で拡大サイズの文字を出し、やはりCHR$(8)で消していきます。

文字を表示する位置は、行番号150と220のXとYの値を変えるだけで、どうにでもなります。

もっと、ゆっくり消していくようにするには、次のようにFOR～NEXTの時間待ちを、行番号185と245で追加しましょう。

```
185 FOR N=1 TO 20:NEXT N （標準サイズをゆっくり消す）
245 FOR N=1 TO 20:NEXT N （拡大サイズをゆっくり消す）
```

タイトルに凝るだけでも、一段とセンスのよいゲームになります。すでに説明ずみの、画面スクロールを利用したタイトルもイケそうですね。

13 着陸船が上昇していく！

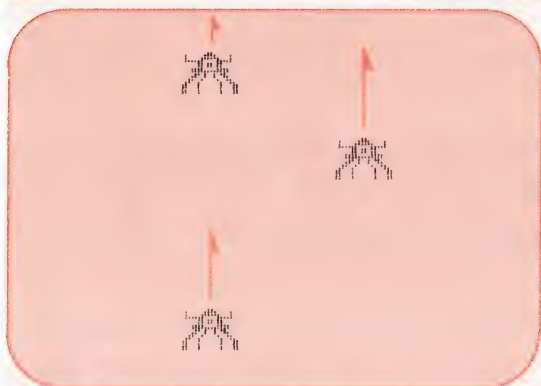
ランディングゲームはご存知ですか？　岩にぶつからないように、逆噴射しながら月面に着陸させる、あれです。またの名を、ムーンランディングとも呼ばれてますね。

ここでは、ランディングゲームの中から、立派な着陸船をつくること、そしてそれをゆっくり上昇させること、をプログラムにしてみました。

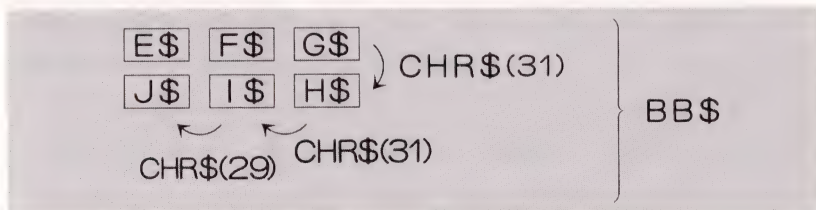
ポイントは、パターン６個でつくった着陸船をいっぺんに動かすことです。

＊6・16　着陸船を動かす

```
100 REM *** 6.16 ***
110 REM *** ﾁｬｸﾘｸｾﾝ ｶﾞ ｳｺﾞｸﾖ ***
120 CLS
130 PATTERNC#224,"0040407C04041C1C"
140 PATTERNC#225,"3078FC84B4B484FC"
150 PATTERNC#226,"001010F08080E0E0"
160 PATTERNC#227,"0C183064C4C4C4C4"
170 PATTERNC#228,"4848840000000000"
180 PATTERNC#229,"C06030988C8C8C8C"
190 E$=CHR$(224):F$=CHR$(225):G$=CHR$(
226):H$=CHR$(227):I$=CHR$(228):J$=CHR$
(229)
200 EE$=CHR$(32):FF$=CHR$(32):GG$=CHR$
(32):HH$=CHR$(32):II$=CHR$(32):JJ$=CHR
$(32)
210 BB$=E$+F$+G$+CHR$(31)+CHR$(29)+CHR
$(29)+CHR$(29)+H$+I$+J$
220 YY$=EE$+FF$+GG$+CHR$(31)+CHR$(29)+
CHR$(29)+CHR$(29)+HH$+II$+JJ$
230 CURSOR18,10:PRINTBB$
240 X=10:Y=20
```

```
250 FOR Y=20 TO 4 STEP -1
260 CURSOR X,Y:PRINTBB$
270 FOR W=0 TO 20:NEXT W
280 CURSOR X,Y:PRINTYY$
290 NEXT Y
```

着陸船が
いくつもいくつも……

プログラムを実行すると、画面のやや右寄りに着陸船を表示し、その左側をもう１つの着陸船がゆっくり上昇していきます。

このプログラムの特徴は、行番号210で、着陸船のパターンを１つにまとめて、ＢＢ＄という変数に代入していることです。そのためには、カーソルを下に移動するコードのCHR$（31)と左へ移動するためのCHR$(29)を利用して、次のような方法でまとめています。

これと同じように、行番号220では、消すための６個のスペースを１つにまとめて、ＹＹ＄という変数に代入しています。

あとは、着陸船ＢＢ＄と表示しては、ＹＹ＄で消していくことを、位置を変えながら行ないます。

14 あみだくじ？ いえ、クレージィ・カーです

画面に格子(こうし)の道路をつくり、その上を自動車が自由自在に動きまわるプログラムをつくってみました。

少々長いプログラムですが、“クレージィ・カー”と名付けました。入力して実行してみてください。

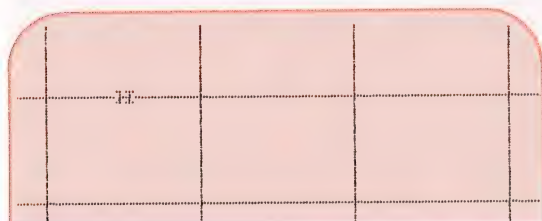
このプログラム、あみだくじにも利用できそうです。

＊6・17　道路を自動車が自由に走り回る

```
10 REM *** 6.17 ***
20 REM *** コノクルマ ドウ ナッテンノ? ***
30 CLS
40 X=8:Y=5:T=1
50 REM
60 FOR Y=2 TO 20
70 CURSORX,Y:PRINT"|"
80 CURSORX+8,Y:PRINT"|"
90 CURSORX+16,Y:PRINT"|"
100 CURSORX+24,Y:PRINT"|"
110 NEXT Y
120 FOR Y=5 TO 15 STEP 5
130 CURSOR 0,Y:PRINT"  ─────────┼─────────┼
─────────┼─────────┼────"
140 NEXT Y
150 REM
160 V=15567
170 VP=VPEEK(V+T)
180 VPOKE V,65
190 REM ヨコ
200 VPOKE V,252
210 FOR W=0 TO 10:NEXT W
220 VP=VPEEK(V+T)
```

```
230 VPOKE V,129
240 IF VP=146 THEN GOTO 370
250 V=V+T
260 IF VP=32 THEN V=V-T:GOTO 490
270 GOTO 190
280 REM タテ
290 VPOKE V,251
300 FOR W=0 TO 10:NEXT W
310 VP=VPEEK(V+L)
320 VPOKE V,128
330 IF VP=146 THEN GOTO 430
340 V=V+L
350 IF VP=32 THEN V=V-L:GOTO 510
360 GOTO 280
370 REM
380 R=INT(RND(1)*3+1)
390 IF R=1 THEN V=V+2:GOTO 190
400 IF R=2 THEN L=40:V=V+L+T:GOTO 280
410 IF R=3 THEN L=-40:V=V+L+T:GOTO280
420 GOTO 190
430 REM
440 R=INT(RND(1)*3+1)
450 IF R=1 THEN V=V+L*2:GOTO 280
460 IF R=2 THEN T=1:V=V+L+T:GOTO 190
470 IF R=3 THEN T=-1:V=V+L+T:GOTO190
480 GOTO 280
490 REM
500 T=T*-1:GOTO 190
510 REM
520 L=L*-1:GOTO 280
```

ウロチョロ
ウロチョロ……

7

音を出す、音を使う

本格派ゲームの総仕上げだ！むろん作曲もＯＫ！！

スイッチオン！　する前に

ピッピッ、ウィ〜ン、ドカ〜ン、ウーウウウ……ゲームを楽しむのに、欠かせないのが、音です。ＵＦＯをうち落とす音、爆発する音など、音がなければ、ゲームの楽しさも半減してしまいますね。この章では、ゲームに必要な効果音や音楽のつくり方を紹介していきます。

1 いろいろな効果音が出せる

ＳＥＧＡ ＳＫ－1100では、いろいろな効果音を出すことができます。

① ＢＥＥＰでピッピッ、SOUNDであれこれ

音を出す方法としては、BEEP(ビープ)とSOUND(サウンド)の２通りがあります。

キーを押したりエラーが発生したときに、ピッとかピポピポという音がしますね。こういう音は、BEEPを使って出せます。それぞれ、次のように指定すると、音の出方が変わってきます。

BEEP	ピッ
BEEP1	ピー
BEEP0	BEEP1の音を止めます
BEEP2	ピポピポ

BEEP音は決まった音しか出せませんが、SOUNDは、いろいろな効果音や音楽をつくり出せます。ＳＫ－1100には、音を出すのに０(ゼロ)～５までのチャンネルがあって、下の周波数や音量をSOUNDを使って指定します。

●チャンネルごとにいろいろな音が出せます

チャンネル	周波数	音量ノイズ	チャンネルの性質
0			音を止める
1	110 ←→ 1760 低い ←→ 高い	0 音を消す 1 ←→ 15 小さい 大きい	音階を出す
2			
3			
4	0~2 3段階の周波数 3 チャンネルで指定する周波数	0 音を消す 1 ←→ 15 こもらない こもる	ホワイトノイズ
5			同期ノイズ

ここからは、SOUNDを使った効果音をいろいろ紹介していきます。プログラム中にある□の部分の数字を変えてみると、効果音が微妙に違ってきます。

まず、次のプログラムを入力してみましょう。

②　ウィ〜ン

＊7・1　「ウィ〜ン」という効果音を出す

```
100 REM *** 7.1 ***
110 REM *** ウィ~ン ウィ~ン ウィ~ン ***
120 FOR I=110 TO 5000
130 SOUND 1,I,15
140 NEXT I
150 SOUND 0
```

実行すると、ウィ〜ンという音が出ます。

③　へんてこな音

＊7・2　へんな音！

```
100 REM *** 7.2 ***
110 REM *** ナンダ゛ コノオトハ? ***
120 FOR I=1 TO 100
130 S=RND(1)*600+200
140 SOUND 1,S,15
150 NEXT I
160 SOUND 0
```

上のプログラムを実行すると、なんだかへんてこな音がします。これは、乱数で周波数を選んでいるからなのです。乱数といえばRNDです。 行番番号130でRND関数を使っています。

④　バクハツ音

＊7・3　ドカーン!というバクハツ音を出す。

```
100 REM *** 7.3 ***
110 REM *** ﾄﾞｶｰﾝ! ﾊﾞｸﾊﾂ ﾀﾞ!! ***
120 FOR I=15 TO 0 STEP -1
130 SOUND 4,1,I
140 FOR T=1 TO 50:NEXT T
150 NEXT I
160 SOUND 0
```

プログラムを実行すると、バクハツ音が出ます。ここで、行番号 130 の□の部分を、0～2の数字に変えて実行してみてください。バクハツ音が、微妙に違ってきます。

⑤　サイレン

＊7・4　ウ〜〜〜〜ウ、パトカーの音!?

サイレ〜ン!

```
100 REM *** 7.4 ***
110 REM *** ｻｲﾚﾝ ﾃﾞｽﾖｵｵｵ ***
120 FOR S=110 TO 900 STEP 5
130 SOUND 1,S,15
140 NEXT S
150 FOR T=1 TO 700:NEXT T
160 FOR S=900 TO 110 STEP -5
170 SOUND 1,S,15
180 NEXT S
190 SOUND 0
```

プログラムを実行すると、ウ〜〜〜〜というサイレン音が出ます。手づくりのゲームの中にパトカーを登場させたいのだったら、この音を使ってみましょう。

⑥　警報音(けいほうおん)

＊　7・5　ウ〰〰、戦闘配置につけ！

```
100 REM *** 7.5 ***
110 REM *** テッキ ライシュウ ***
120 FOR I=1 TO 30
130 FOR S=1000 TO 900 STEP -5
140 SOUND 1,S,15
150 NEXT S
160 FOR S=900 TO 1000 STEP 5
170 SOUND 1,S,15
180 NEXT S
190 NEXT I
```

プログラムを実行すると、ウ〰〰〰という音がしますが、⑤のサイレン音とはちょっと違いますね。こちらは、「戦闘配置につけ！」のときに鳴る警報音です。このプログラムの行番号130と160にあるステップ数（－5と5）を小さくすると、警報音の鳴り方が速くなります。

⑦　電話のベル

＊7・6　リ～ン、リ～ン、電話ですよ！

```
100 REM *** 7.6 ***
110 REM *** リ~ン リ~ン リ~ン ***
120 FOR I=1 TO 30
130 FOR M=1 TO 30
140 SOUND 3,1000,15
150 FOR T=1 TO 2:NEXT T
160 SOUND 3,1000,0
170 NEXT M
180 FOR T=1 TO 200:NEXT T
190 NEXT I
```

プログラムを実行すると、リリリーン、リリリーンという電話のベルの音が出ます。□の部分の数を変えて、あなたの家の電話のベルに近い音にしてみましょう。

⑧　お話中の音

＊7・7　ツーツーツー……

```
100 REM *** 7.7 ***
110 REM *** ﾀﾀﾞｲﾏ ｵﾊﾅｼﾁｭｳ ﾃﾞｽ ***
120 FOR I=1 TO 30
130 SOUND 1,400,15
140 FOR T=1 TO 200:NEXT T
150 SOUND 0
160 FOR T=1 TO 200:NEXT T
170 NEXT I
180 SOUND 0
```

プログラムを実行すると、ツーツーという電話のお話中の音がします。よく聞くと、音が鳴っている時間と、音が止まっている時間は同じ長さですね。これは、行番号140と160で、同じ長さを指定しているからです。

⑨　爆弾投下の音

＊7・8　ヒューン、ドカン！爆弾だ!!

```
100 REM *** 7.8 ***
110 REM *** ﾊﾞｸﾀﾞﾝ ﾄｳｶ! ***
120 FOR S=900 TO 500 STEP -5
130 SOUND 1,S,15
140 NEXT S
150 SOUND 0
160 FOR I=15 TO 0 STEP -1
170 SOUND 4,1,I
180 FOR T=1 TO 50:NEXT T
190 NEXT I
```

プログラムを実行すると、爆弾が投下されて爆発したような、ヒューン、ドカン！　という音がします

⑩　ピュンピュン

＊7・9　ミサイル発射する！

```
100 REM *** 7.9 ***
110 REM *** ミサイル ハッシャ ノ オトダ゛ヨ ***
120 FOR M=1 TO 10
130 FOR S=100 TO 0 STEP -10
140 SOUND 3,S+900,S*0.15
150 NEXT S
160 SOUND 0
170 FOR T=1 TO 200:NEXT T
180 NEXT M
```

プログラムを実行すると、ピュンピュンという、ミサイルを発射するような音がします。この「ピュン」という音は、音程と音量を同時に下げることによって発生します。行番号120のステップを小さくしてみましょう。音が長くなって、「ピューン」となるはずです。

⑪　ドシンドシン

＊7・10　モンスターが歩く音を出す

```
100 REM *** 7.10 ***
110 REM *** カイシ゛ュウ カ゛ アルク オトダ゛ヨ ***
120 FOR M=1 TO 10
130 FOR S=100 TO 0 STEP -10
140 SOUND 3,S+900,S*0
150 SOUND 4,3,S*0.15
160 NEXT S
170 SOUND 0
180 FOR T=1 TO 200:NEXT T
190 NEXT M
```

プログラムを実行すると、ドシンドシンという、モンスターが歩くような音が出ます。行番号150のチャンネル4を5にして、同期ノイズにすると、宇宙的でおもしろい音になりますよ。

CAR!!

⑫ 自動車が通過する音

＊7・11 自動車が近づき、そして遠ざかる

```
1 REM *** 7.11 ***
5 REM *** クルマ ノ エンジ゛ン ノ オトダ゛ヨ ***
10 FOR I=1500 TO 3000 STEP 10
20 SOUND 3,I,0
30 SOUND 5,3,15-ABS(I/100-20)
40 NEXT I
50 SOUND 0
```

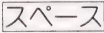
プログラムを実行すると、自動車が遠くから近づいてきて、また遠ざかっていくような音が出ます。ゲームに利用できそうな音ですね。

⑬ こんな効果音は？

せっかく効果音が出せるのですから、キャラクタの動きに合った効果音をつくってみましょう。

次のプログラムは、砲台を動かしてミサイルを発射し、標的を撃つというプログラムに、効果音をつけ加えたものです。

＊7・12 ミサイル発射、標的を撃て!

```
100 REM *** 7.12 ***
110 REM *** センソウ エイガ゛ ミタイダ゛! ***
120 CLS:CURSOR 2,10:PRINT"OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO"
130 X=20
140 GOTO 200
150 A$=INKEY$:IF A$=" " THEN 230
160 IF A$=CHR$(28) THEN X=X+1
170 IF A$=CHR$(29) THEN X=X-1
```

```
180 IF X<2 THEN X=2:BEEP
190 IF X>34 THEN X=34:BEEP
200 CURSOR X-1,22:PRINT" ↑ ";
210 CURSOR X-2,23:PRINT" ▲ ";
220 GOTO 150
230 Y=22:C=32:S=1000
240 VPOKE 15362+X+Y*40,32
250 IF C<>32 THEN 310
260 S=S-10:SOUND 1,S,10
270 Y=Y-1:A=15362+X+Y*40
280 C=VPEEK(A):VPOKE A,142
290 IF Y<0 THEN 400
300 GOTO 240
310 SOUND 0
320 FOR I=15 TO 0 STEP -1
330 SOUND 4,0,I:FOR T=1 TO 5:NEXT T
340 NEXT I
350 GOTO 200
400 SOUND 5,2,15
410 FOR T=1 TO 100:NEXT T
420 SOUND 0:GOTO 200
```

プログラムを実行すると、砲台と標的を表示します。砲台は、[←][→]を使って左右に動かすことができます。[スペース]を押すと、ミサイルを発射します。ミサイルが標的に当たると、標的は消えます。これらの動きに合わせて、次のような効果音が出ます。

ミサイル発射………ヒューン
当たり………………バーン　　　はずれ…………ブー
砲台が端に…………ピッピッピッ

2 メロディを楽しもう

SOUNDを使うと、効果音だけでなく、メロディをつくり出すことだってできます。ちょっと才能があれば、あなたもシンガーソングライターになれるかも？

① ドレミ音

次の表のように、決まった周波数（単位はHz）を指定すれば、音階が出せます。音階を出すときには、チャンネル1から2を使います。

●SK1100では周波数を指定する

音階			+1	+2	+3	+4	+5
C	ド	ハ		131	262	523	1047
$C^{\#}$・D^{b}				139	277	554	1109
D	レ	ニ		147	294	587	1175
$D^{\#}$・E^{b}				156	311	622	1245
E	ミ	ホ		165	330	659	1319
F	ファ	ヘ		175	349	698	1397
$F^{\#}$・G^{b}				185	370	740	1480
G	ソ	ト		196	392	784	1568
$G^{\#}$・A^{b}				208	415	831	1661
A	ラ	イ	110	230	440	880	1760
$A^{\#}$・B^{b}			117	233	466	932	
B	シ	ロ	123	247	494	988	

この表から▬▬の部分の周波数をひろって、ドレミファソラシドを出してみましょう。

＊7・13　ドレミの音を出してみよう

```
1 REM *** 7.13 ***
5 REM *** ト゛レミファソラシト゛ ダ゛ヨ ***
10 READ A
20 IF A<0 THEN END
30 SOUND 1,A,15
35 FOR T=1 TO 50 :NEXT T
36 SOUND 0
37 GOTO 10
40 DATA 131,147,165,175,196,220,247,26
2
50 DATA -1
```

②　流れるメロディ

＊7・14　なんのメロディでしょう？

```
1 REM *** 7.14 ***
5 REM *** ヘ゛ートーヘ゛ン ノ キョク テ゛スヨ ***
10 RESTORE 80
20 READ D
30 IF D=0 THEN SOUND0:END
40 SOUND 1,D,15
50 SOUND 2,D*2,11
60 SOUND 3,D*3,7
65 FOR W=0 TO 50:NEXT W:REM オトノ ナカ゛サ
70 GOTO 20
80 DATA 370,370,392
81 DATA 440,440,392
82 DATA 370,330,294
83 DATA 294,330,374
84 DATA 370,330,330
85 DATA 370,370,392
86 DATA 440,440,392
87 DATA 370,330,294
88 DATA 294,330,370
89 DATA 330,294,294,0
```

③　キーボードがオルガンに変身

ちょっと長いですが、次のプログラムを入力してみましょう。

＊7・15　オルガン演奏のプログラム

```
10 REM *** 7.15 ***
15 REM *** ｵﾙｶﾞﾝ ﾉ ｵﾄ ｶﾞ ﾃﾞﾙﾖ ***
20 CLS:GOSUB 250
30 Z$=INKEY$
40 IFZ$="A"THEN SOUND1,220,15:GOTO200
45 IFZ$="W"THEN SOUND1,233,15:GOTO200
50 IFZ$="S"THEN SOUND1,247,15:GOTO200
60 IFZ$="D"THEN SOUND1,262,15:GOTO200
65 IFZ$="R"THEN SOUND1,277,15:GOTO200
70 IFZ$="F"THEN SOUND1,294,15:GOTO200
75 IFZ$="T"THEN SOUND1,311,15:GOTO200
80 IFZ$="G"THEN SOUND1,330,15:GOTO200
90 IFZ$="H"THEN SOUND1,349,15:GOTO200
95 IFZ$="U"THEN SOUND1,370,15:GOTO200
100 IFZ$="J"THENSOUND1,392,15:GOTO200
105 IFZ$="I"THENSOUND1,415,15:GOTO200
110 IFZ$="K"THENSOUND1,440,15:GOTO200
115 IFZ$="O"THENSOUND1,466,15:GOTO200
120 IFZ$="L"THENSOUND1,494,15:GOTO200
130 IFZ$=";"THENSOUND1,523,15:GOTO200
135 IFZ$="@"THENSOUND1,554,15:GOTO200
140 IFZ$=":"THENSOUND1,587,15:GOTO200
145 IFZ$="["THENSOUND1,622,15:GOTO200
150 IFZ$="]"THENSOUND1,659,15:GOTO200
200 IF Z$="" THEN SOUND 0
210 GOTO 30
250 PRINT" #ﾗ  #ﾄﾞﾚ  #ﾌｧｿ ﾗ #ﾄﾞﾚ "
260 PRINT"  W   R T   U I O  @ [ "
270 PRINT
280 PRINT" A S D F G H J K L ; : ]"
290 PRINT" ﾗ ｼ ﾄﾞﾚ ﾐ ﾌｧｿ ﾗ ｼ ﾄﾞﾚ ﾐ"
300 RETURN
```

プログラムを実行すると、画面に音階とそれに対応するキーを表示しま

す。出したい音に対応するキーを押してください。たとえば、〝ファ〞の音を出したいときには、[H]を押します。キーボードを、オルガンのように使って、簡単な音楽が演奏できますね。

④　作曲しよう

さて、いよいよ音の最終段階、作曲です。次のプログラムを入力してみましょう。

＊7・16　自分で作曲しよう

```
10 REM *** 7.16 ***
20 REM *** キョクメイ ヲ アテテ ミヨウ ***
30 TE=1000:L=4
40 READ A
50 IF A<0 THEN 130
60 V=15
70 IF A=1 THEN V=0:A=110
80 IF A<110 THEN L=A: GOTO 40
90 SOUND 0
100 SOUND 1,A,V
```

```
110 FOR T=1 TO TE/L:NEXT T
120 GOTO 40
130 SOUND 0
1000 DATA 16,784,659,659,784,784,659,6
4,880,784,32,698,16,784,32,698,16,784,
659
1010 DATA 1
1020 DATA 16,784,659,659,784,784,659,6
4,880,784,32,698,16,784,659,659,784,78
4,8
1030 DATA 659,1
1040 DATA 16,784,880,698,64,880,784,16
,698,2,1047,-1
```

プログラムを実行すると、作曲したメロディを演奏します。行番号1000から後のデータを書き直せば、あなたの作曲したメロディが流れます。

実際に作曲して、データを指定するには、次のようにします。

●作曲するときは指定を間違えずに

110以上の数	ドレミの音階(①ドレミ音参照)
0未満	データの終わり
1	休符
2~110	音符の長さ 2=2分音符(𝅗𝅥) 4=4分音符(♩) 8=8分音符(♪)

単に音を並べるだけでなく、音量を変えてみたり、くり返しを指定したりして、あなたのオリジナルサウンドをつくってみてください。きっと楽しいものができ上がると思いますよ。

8

基本プログラミングのコツ

いよいよゲームづくりの総まとめ！もちろんアニメもネ！

スイッチオン！　する前に

今までお話ししたものをうまくまとめると、ゲームがつくれます。つまり、１章から７章までは、ゲームの部品のようなかたちでしたが、ここでは、それを組み合わせて、ちゃんとしたゲームをつくります。

さあ、どんなゲームができ上がるか、中には、点をとるのが意外とむずかしい「テレビテニス」もありますよ。

1 サイコロとルーレットはこんなに簡単

先が見えない、何が出てくるかわからない、というのがゲームのおもしろさです。さて、ゲームには、必ず、この予測できない情況をつくるための命令が使われています。まずは、そのことについて、ちょっとお話ししておきましょう。

それは、RND関数（ランダムかんすう）といって、乱数（らんすう）を発生させる働きをもつものです。

乱数とは、でたらめな数のこと、乱数発生とは、でたらめな数を出すことです。つまりRND関数を使うと、コンピュータにでたらめな数を出させることができるのです。論より証拠。?RND(1)と入力し、RETURNキーを押してみましょう。続けて何度も?RND(1) RETURNをくり返してみましょう。出てくる数は、まちまちで、1つとして同じ数は出ませんね。これを利用すると、ゲームができます。

●サイコロゲーム

たとえば、サイコロ。まあゲームというにはちょっとモノ足りませんが、乱数発生（RND関数）を使ってみるのには最適でしょう。ウォーミング・アップのつもりで、次のプログラムを実行してみます。

＊8・1　1から6までの数字を出す

```
10 REM *** 8.1 ***
20 REM *** ｻｲｺﾛｹﾞｰﾑ ***
30 R=INT(RND(1)*6)+1
40 PRINT R;
50 A$=INKEY$:IF A$="" THEN 50
60 GOTO 30
```

実行すると、１から６までの中から、ある数字を１個だけ表示します。そして、何でもいいですからキーを押すと、また１から６までの中から、数字を表示します。キーを押し続けていると、１から６までの範囲の中で、でたらめな数を次々と表示していきます。

この１から６までの数字がどうしたの？　なんて言わないこと。これがちょうどサイコロの目にあたります。サイコロを振ったときに、どの目が出るか、その結果は予測できないものですね。この、振ってみなければわからない、といったことに乱数発生（RND関数）を使っています。

ところが、このRND関数、ちとクセがありまして、そのまますなおに使うわけにはいきません。行番号３０を見てください。ＩＮＴ関数と組み合わせたり、６をかけたり、さらに１をたしたりというように、かなり工夫をしてRND関数を使っていることがわかります。

そこで、そのワケを、次に説明しておきましょう。

もともとRND（１）という関数は、０から１までの乱数を発生するものです。もっと厳密に言うと、０以上で１未満の数、すなわち０と１は含まない０から１までの数を、まんべんなく生み出すことになります。

ということは、発生する値は整数ではなく、０・２３８０５４６２９４などの使いにくい数です。

そこで、この扱いにくい数字を整数、つまり１とか２、０、－１－２…というようなきりのいい数字にするために、INT関数を組み合わせています。ただし、単にINT関数を組み合わせただけではダメで、６をかけてからにします。そうすると、０、１、２、３、４、５という整数が発生するようになります。

しかし、サイコロの目は１から６までに決まっていますので、１をたすことによって、１、２、３、４、５、６が発生するようにしています。

こうして、行番号30のINT(RND(１)＊６)＋１となるのです。

それでは、０から９の乱数を発生させるためにはどうしたらいいか、こ

れは簡単、行番号30を次のように変えるだけです。

30　R＝INT（RND（1）＊10）　………0～9の乱数

そして、1から10までの乱数にするには、これに1を加えるだけです。

30　R＝INT（RND（1）＊10）＋1　………1～10の乱数

この調子で、1から100でも、1から1000でも、どのような範囲の乱数も出すことができます。

●ルーレットに挑戦(ちょうせん)！

こんどは、ルーレットに挑戦してみましょう。

ルーレットには、0(ゼロ)から36までの数と、00という数（？）があります。この中のどれかが出るように、乱数発生を工夫します。

0から36までの数は、これまでの方法と同じですが、それでは00はどうしたらいいのでしょうか。いたってカンタンです。0から37の数を発生させて、37と出たときには〝00〟を表示するようなプログラムにすればいいのです。そうしてつくったプログラムが、次のものです。

＊8・2　ルーレットのルールづくり

```
10 REM *** 8.2 ***
20 REM *** ルーレットゲーム ***
30 R=INT(RND(1)*38)
40 IF R=37 THEN 80
50 PRINT R
60 A$=INKEY$:IF A$="" THEN 60
70 GOTO 30
80 PRINT" 00"
90 GOTO 60
```

実行すると、00、0から36までの数のうち、いずれかの数が出ます。適当なキーを押すごとに、どれかの数字が出ます。

実行すると……

```
RUN
 3
 34
 18
 28
 11
  5
 29
Break in 60
```

「なあんだ、ちっともサイコロらしくない、ルーレットらしくない」と、不満を抱いている人もいるでしょう。その気持ちは、とってもよくわかります。それでは、どうすればそれらしくなるのでしょうか。

たとえば、サイコロの場合、１から６までの数字が出るだけではおもしろくありません。そこでサイコロの絵が出るようにすると、かなりそれらしくなります。

具体的には、⚀、⚁、⚂、⚃、⚄、⚅というサイコロのパターンをつくり、数字の代わりに表示させるようにします。ON～GOSUB や IF～THEN をうまく使ってやってみてください。

また、ルーレットの場合も、同じように画面上に絵が出るようにすると、かなり楽しいものになります。これまでにやってきたアニメーションをとり入れたり、音を使ったりすることで、かなり本格的になりそうです。

つまり、ゲームの決め手は、どのように画面をつくるか、どんな絵をとり入れるか、にあるのです。ルールの部分は、わりと短いプログラムで済むのですが、ゲームのプログラムが長くなってしまうワケは、それらしい絵を表示させたり、動きをとり入れたりしているためなのです。

ひとつ、お父さんと一緒にプログラムづくりを競争しながらやってみて、実力をためしてみてはいかが？

2 ルールのつくり方のコツ

ゲームの基本はルールにあります。ルールがたくさんあればあるほど、高度なゲームです。たとえばマージャン。家族そろって楽しんでいる人もいるようですが、さてこのルールの部分は、どんなふうにつくったらいいでしょうか。ここでは、ルールづくりのコツをつかむために、フクビキを例にとってプログラムにしてみましょう。

まず、フクビキのルールを決めます。

実際のフクビキは、決まった数の玉が入っていて、1等は1個、2等は3個、3等は10個、……というように、当たりの数も決められています。これとまったく同じようにプログラムをつくることもできますが、もっと簡単なやり方があります。それは、確率(かくりつ)という考え方を使った方法で、100回に1度は1等に、50回に1度は2等になるといったことです。

そこで、フクビキのルールは、次のように決めたとします。

1等賞	100回に1度の割合で当たる	確率1/100
2等賞	50回に1度の割合で当たる	確率1/50
3等賞	10回に1度の割合で当たる	確率1/10
4等賞	5回に1度の割合で当たる	確率1/5
ハズレ	その他全部	

これを乱数で表わすのは、思いのほか簡単です。RND(1)とすると、0(ゼロ)以上1未満の数がまんべんなく発生するのですから、1/100つまり0.01以下の数が発生する確率は1/100とみることができます。同じように1/50 0.02以下の数が発生する確率も、1/50です。同様に1/10の0.1以下が発生す

る確率は1/10であり、1/5の0.2以下の数が発生する確率は1/5です。

これを１等から順番に並べると、次のようなプログラムができます。

＊8・3　フクビキはこうしてつくる

```
10 REM *** 8.3 ***
20 REM *** フクビキゲーム ***
30 PRINT:FOR T=1 TO 100:NEXT T
40 A$=INKEY$:IF A$="" THEN 40
50 R=RND(1)
60 IF R<1/100 THEN 120
70 IF R<1/50 THEN 140
80 IF R<1/10 THEN 160
90 IF R<1/5 THEN 180
100 PRINT"---- ザンネン ハズレ ----"
110 GOTO 30
120 PRINT"---- オメデトウ 1トウショウ ----"
130 GOTO 30
140 PRINT"---- アタリ! 2トウショウ ----"
150 GOTO 30
160 PRINT"---- 3トウショウ アタリ ----"
170 GOTO 30
180 PRINT"---- 4トウ ザンネンショウ ----"
190 GOTO 30
```

プログラムを実行し、適当なキーを押します。そうすると、１等、２等、３等、４等、ハズレのうち、いずれかを表示します。なお、ハズレについては１等～４等以外のものということですから、IF～THENを使わなくてもいいわけです。

プログラムをつくるときに注意することがあります。いくつかの判定を並べていく場合は、必ず確率の低いものから順に書かねばなりません。もし、４等賞を先にしてしまうと、１等、２等、３等は、絶対に当たらないフクビキになってしまいますよ。

3 ジャンケンゲームで頭をならそう

乱数発生のしめくくりに、ジャンケンゲームをつくりました。どんなものか、ひととおり試してみましょう。

＊8・4　ジャンケンゲームはこのプログラムでOK

```
10 REM *** 8.4 ***
20 REM *** ジャンケンゲーム ***
30 CLS:C=0:Y=0:M=0
40 R=INT(RND(1)*3)+1
50 PRINT:INPUT"ナニヲ ダシマスカ?";A
60 C=C+1
70 PRINT"アナタ ハ  ";:S=A:GOSUB 240
80 PRINT"ワタシ ハ  ";:S=R:GOSUB 240
90 IF A=R THEN 160
95 ON R GOTO 100,120,140
100 IF A=3 THEN 180
110 GOTO 200
120 IF A=1 THEN 180
130 GOTO 200
140 IF A=2 THEN 180
150 GOTO 200
160 PRINT"ヒキワケ デス "
170 GOTO 220
180 PRINT"アナタ ノ カチ!!!"
190 Y=Y+1:GOTO 220
200 PRINT"ワタシ ノ カチ!!!"
210 M=M+1:GOTO 220
220 PRINT"アナタ ノ ショウリツ ......";Y/C
230 GOTO 40
240 IF S=1 THEN PRINT"グー"
250 IF S=2 THEN PRINT"チョキ"
260 IF S=3 THEN PRINT"パー"
270 RETURN
```

実行すると、〝ナニヲ　ダシマスカ？〟と問いかけてきますので、グーを出す場合は１を、チョキなら２を、そしてパーなら３を、入力します。そうすると、ＳＫ－1100が出したものと判定して、〝アナタ　ノ　カチ〟〝ワタシ　ノ　カチ〟〝ヒキワケ　デス〟のいずれかを表示し、さらに勝率も教えてくれます。

さて、ここでは、プログラムの全体の構成について、ちょっと考えてみたいと思います。

まず、行番号30、ここでは、主にプログラムの中で使っていく変数をはっきりと示しています。というのも、変数というのは便利なようで、わかりにくいものです。意味のある整数は、最初に宣言しておくと、あとで混乱することもなくなります。この部分を、カッコよく「初期設定（しょきせってい）」と呼びます。

行番号40から80までは、ジャンケン、ポンの部分です。

行番号90から230までは、勝負の判定をしています。

行番号240以降は、〝グー〟〝チョキ〟〝パー〟を、画面に表示するための部分です。

このように、どこで何をしているかがはっきりわかるように、プログラムをつくっていきます。そうすることで、修正するのも簡単になり、また、プログラムの中の１部分を他のプログラムに利用することもできるからです。

コンピュータと
ジャンケン！

```
RUN

ﾅﾆｦ ﾀﾞｼﾏｽｶ?1
ｱﾅﾀ ﾊ  ｸﾞｰ
ﾜﾀｼ ﾊ  ｸﾞｰ
ﾋｷﾜｹ ﾃﾞｽ
ｱﾅﾀ ﾉ ｼｮｳﾘﾂ ...... .0833333

ﾅﾆｦ ﾀﾞｼﾏｽｶ?
```

4 いよいよ テレビテニスに挑戦

お待ちどうさま。いよいよゲームらしいゲーム、動きのあるゲームをつくるコーナーに到着しました。

これからつくろうとするテレビテニスは、テレビゲームの中では最も古く、かれこれ10年以上も前からあるものです。今や、ゲームセンターのゲーム機も最新鋭のものばかりで、テレビテニスにお目にかかることもなくなりましたが、反射神経を競うゲームでは、ブロックくずしよりも前に、人気があったものです。

さて、このゲームがどのようなものかを、知らない人のために説明しておきましょう。

まず、画面上にテニスコートが現われます。テニスコートの両端に、ラケットを表示します。このラケットを動かして、飛んできたボールに当てます。そうすると、ボールがはね返り、相手のコートに飛んでいきます。相手側でもラケットを動かして、ボールを打ち返します。

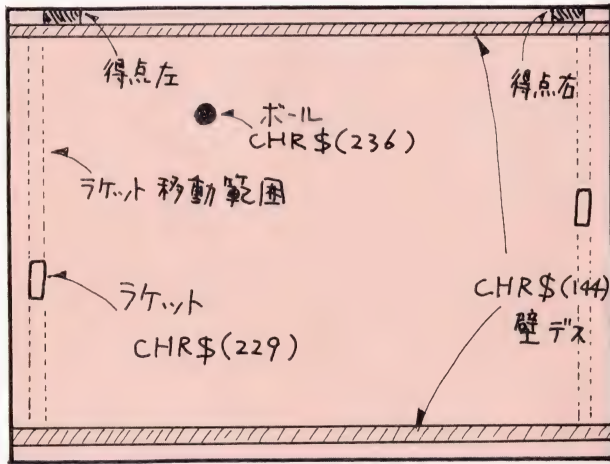
つまり、ラケットにうまくボールを当てて、やりとりするゲームです。はずれてしまうと、相手の得点となります。

それでは、ゲームづくりに入りましょう。

● 〈画面の設計をします〉

こういったゲームの場合、まず最初にしなければならないのは、画面の設計です。画面の設計とは、どういった画面にするかを決めてしまうことで、そのためには、画面上のどこに何を表示するのかを、はっきりさせてしまいます。

画面設計の方法は、方眼紙などを使って、画面の大きさ分（たとえばキャラクタ画面の場合は、タテ24×ヨコ38)のマス目をつくり、それを画面にみたてて設計します。

そうして設計したテレビテニスの画面は、次のようになります。

ここでは、画面設計のついでに、表示するキャラクタも決めてしまいました。テニスコートを表わす画面の上下の太い線(壁)は、キャラクタコードの144です。このコードは、そこにボールが当たったときにはね返るようにするときに使います。ラケットは、キャラクタコードの229を2つつないでつくります。ボールは、キャラクタコードの236です。

● 〈プログラムを組みます〉

次に、でき上がったプログラムを見てみましょう。

実際に自分でイチからつくるときは、先にプログラムを見るわけにはいきませんが、ここではプログラムを見ておいたほうがわかりやすくなります。

＊8・5　テレビテニスのプログラム

```
100 REM *** 8.5 ***
105 REM *** ﾃﾚﾋﾞ ﾃﾆｽ ***
110 CLS
120 SL=0:SR=0:PL=10:PR=10:V=15362:VL=V
+2:VR=V+35:X=0:Y=2:DX=1:DY=1
130 CURSOR0,1:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
140 CURSOR0,22:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
160 A$=INKEY$:IF A$="" THEN 215
165 C=ASC(A$)
170 IF C=30 THEN GOSUB1000
180 IF C=31 THEN GOSUB1100
190 IF C=81 THEN GOSUB1200
200 IF C=90 THEN GOSUB1300
210 IF C=8 THEN BF=1
215 IF BF=0 THEN 160
220 NX=X+DX:NY=Y+DY
230 IF NX<0THEN SR=SR+1:SF=0:GOTO2000
240 IF NX>37THEN SL=SL+1:SF=1:GOTO2000
250 P=VPEEK(V+NX+NY*40):IF P=32 THEN50
0
260 IF P=144 THEN DY=-DY:GOTO400
270 IF P=229 THEN DX=-DX:GOTO400
400 GOTO 160
500 VPOKE V+X+Y*40,32:X=NX:Y=NY
510 VPOKE V+X+Y*40,236
520 GOTO160
1000 PR=PR-1:IF PR<2 THEN PR=2:RETURN
1010 VPOKE PR*40+VR,229
1020 VPOKE (PR+2)*40+VR,32:RETURN
1100 PR=PR+1:IF PR>20 THEN PR=20:RETUR
N
1110 VPOKE (PR-1)*40+VR,32
1120 VPOKE (PR+1)*40+VR,229:RETURN
1200 PL=PL-1:IF PL<2 THEN PL=2:RETURN
```

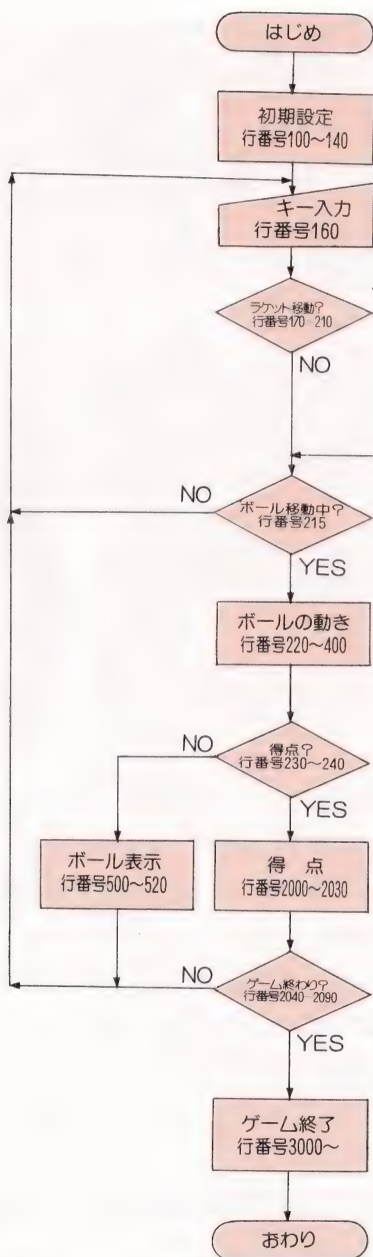
```
1210 VPOKE PL*40+VL,229
1220 VPOKE (PL+2)*40+VL,32:RETURN
1300 PL=PL+1:IF PL>20 THEN PL=20:RETUR
N
1310 VPOKE (PL-1)*40+VL,32
1320 VPOKE (PL+1)*40+VL,229:RETURN
2000 VPOKE V+X+Y*40,32:BF=0
2010 CURSOR 1,0:PRINTSL:CURSOR 34,0:PR
INTSR
2020 IF SF=1 THEN X=0:Y=INT(RND(1)*20)
+2:DX=1:DY=1
2030 IF SF=0 THEN X=37:Y=INT(RND(1)*20
)+2:DX=-1:DY=1
2040 IF SL=15 THEN W$="ﾋﾀﾞﾘ":GOTO3000
2050 IF SR=15 THEN W$="ﾐｷﾞ":GOTO3000
2090 GOTO160
3000 CURSOR 15,8:PRINT"GAME OVER"
3010 CURSOR 15,12:PRINTW$;" ﾉ ｶﾁ"
3020 CURSOR 14,15:INPUT"PLAY AGAIN?";A
$
3030 IF A$="Y" THEN 110
```

● 〈全体の構成と処理の流れを見てみましょう〉

プログラムを分けてみると、初期設定からゲーム終了まで、大きく7つの部分に分かれます。そして、その中を行ったりきたりすることでゲームが成り立っています。こういった処理の流れをフローチャート（流れ図）にしてみると、次ページのようになります。

こういった反射神経を競うゲームでは、〝キー入力〟と〝キャラクタの移動〟の繰り返しが基本です。このパターンは、ブロックくずし、インベーダーゲームなどでも同じようなものです。

● 〈プログラムの説明〉

全体をながめたところで、こんどは、プログラムの各部分について見ていきましょう。

フローチャートは、これを見ただけで頭が痛くなるという人がたくさんいます。反対に、これがなければプログラムが組めないという人もいます。ですから、あってもなくても、どちらでもいいのです。ただ全体の流れをつかむには、それなりに便利な図です。もし、あなたが、「フローチャートがあってもいいな」と思ったなら、プログラムをつくるときに、自分なりに書いてみてください。書き方はどうでもよく、矢印と、判断のＹｅｓ、Ｎｏとがあれば、立派なフローチャートができあがります。

初期設定では、行番号120で変数の準備を、130と140でテニスコートの壁の表示を行ないます。ところで、行番号120でわかるように、このプログラムではずいぶんと多くの変数を使っています。変数が多くなると、それぞれの変数の意味もわかりにくくなってしまいます。そのような場合には、変数リストというものをつくっておくと便利です。

●テニスゲームの変数リスト

変数	意　　味	最初の値
SL	左側の得点	0
SR	右側の得点	0
PL	左ラケット表示位置(X座標)	10
PR	右ラケット表示位置(X座標)	10
V	画面の先頭アドレス	15362
VL	左ラケット位置アドレス	V+2
VR	右ラケット位置アドレス	V+35
X	ボールのX座標	0
Y	ボールのY座標	2
Dx	ボールの移動方向(X方向)	1
Dy	ボールの移動方向(Y方向)	1

次の行番号160からはキー入力の部分です。ここでは、右ラケットを上下に移動するための↑、↓、左ラケットの上下移動にQ、Z、そして、ボールをサーブするために DEL 、といったキー操作のところをプログラムにしています。

もっと細かく説明すると、行番号160がキー入力で、165では押されたキーのコードを変数Cに入れています。170から210までの部分では、それぞれのキーに応じた働きをするように判断させています。

その中の行番号210については、ちょっと説明が必要ですね。ここでは、ボールをサーブするための DEL が押されたときにどうするかを判断さ

せていますが、ただBFという変数に1を代入しているだけです。とはいっても、BF=1としただけでボールがサーブできるわけではなく、その後で行番号220以降に進み、そこでボールをサーブするようになっているのです。

それでは何のために、わざわざBF=1という代入文を書いているのでしょうか。それは、こういうことなのです。

変数BFには、0か1のどちらかの値が入るようになっています。BFが0のときには、ボールは画面上にはなく、サーブの準備状態にあります。BFが1のときには、ボールが飛んでいる途中にあります。つまり、変数BFは、ボールが飛んでいるかどうかを示すための変数です。こういった変数を、フラグ(旗)と呼び、スイッチ的な役割を果たします。

というわけで、行番号210では、DEL が押されると、BF=1として、ボールが飛んでいる状態にフラグをセットしているのです。そして、行番号215で、変数BFの中身を確認し、0のときは160へ戻り、0でないとき、つまりBF=1のときは、220へ進みます。

BF=0というのは、サーブ前に DEL 以外のキーを押したときの状態です。つまり、まだサーブは行なわれていないため、ボールの動きは不必要です。そこで行番号160へ戻っているのです。BF=1の場合のみ、行番号220へ進んでいます。

行番号220から400のボールの動きについては、110ページで説明したとおりです。ただ、行番号230と240では、SR、SLという2つの変数を使って得点の計算も行なっています。また、変数SFをフラグとして

使っていますが、これはもう少し後で説明します。

ボールの移動表示は、行番号500から520です。これについては、今さら説明の必要もないでしょう。位置を変えながら、書いて消しての繰り返しです。

行番号1000から1320のラケット移動についても、106ページで説明したとおりです。

さて、行番号2000から2090の得点の処理について、ちょっと説明しておきましょう。

得点そのものの計算は、すでに行番号230と240でやっていますので、ここでは、得点の表示、勝負の判定、そしてサーブの準備をしています。行番号2000では、得点までいったボールを消して、BF＝0として、ボールが画面上に出ていない状態にセットしています。そして、行番号2010で、得点を表示します。

行番号2020と2030では、フラグとして使うもうひとつの変数SFが登場してきます。ここでSFの意味を説明しておきましょう。

SFは、左右どちらがサーブをするのかを示すフラグです。SF＝0(ゼロ)なら右側のサーブ、SF＝1ならば左側のサーブです。つまりサーブ権の移行に、フラグSFを使っているわけです。サーブ権は得点となった方が獲得するものです。その準備を、行番号2020と2030で行なっています。

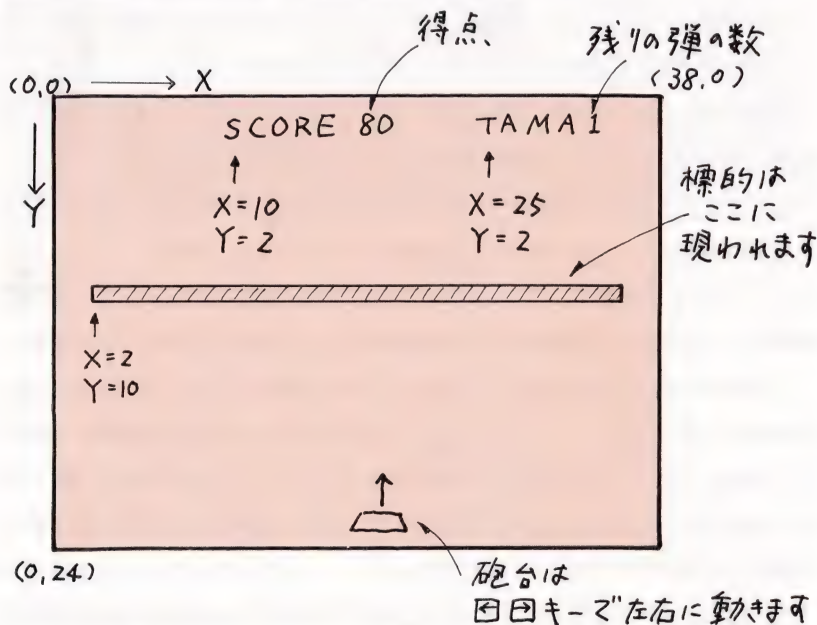
行番号2040と2050はマッチポイントの判定です。このゲームでは、15点をマッチとしましたので、SLかSRのどちらかが15になれば、試合終了です。SL＝15またはSR＝15となったところで、変数W$に勝ったほうを入れて、行番号3000へ飛びます。SL、SR共に15でなければ、行番号160に戻って試合を続けます。

行番号3000からは、“GAME OVER” の表示と、勝ったほうの表示をしています。その後、もう一度ゲームをするかどうかをきき、そこでYと入力すると、再びゲームを楽しむことができます。

5 おつぎは撃墜(げきつい)ゲームの巻

119ページのプログラムをグレードアップして、撃墜ゲームに仕立て上げましょう。次のページのプログラムリストを見てください。

それでは、この「撃墜ゲーム」の内容はどうなっているのか、みていきましょう。

まず画面設計です。このプログラムを実行すると、画面には下のような砲台(ほうだい)、標的(ひょうてき)、SCORE、TAMAを表示します。

砲台の位置は←→キーで自由に移動することができます。弾の発射はスペースキーです。スペースキーを押すと↑が上へ飛んでいきます。標的は

“※”“O”“☻”の３種類が次々と現われ、右から左へ流れていきます。弾は１ゲーム20発まで射つことができ、射った残りの数を“TAMA”のところに表示します。

得点は、砲台の位置によって違います。弾の流れが右から左ですので、砲台は左にあったほうが的中率は高くなります。そこで砲台が左にあるほど得点は低くしておきます。また当たった標的によっても得点が違います。“O”と“※”はプラスの得点、“☻”とミスショットはマイナスの得点となります。

プログラム中では、変数ＴＡが弾の数、ＳＣが得点、Ｘが砲台のヨコ方向位置、Ｙが弾のタテ方向位置になっています。行番号1000からと行番号2000からは、それぞれ標的と得点の表示を行なっています。

＊8・6　撃墜ゲームのプログラム

```
5 REM *** 8.6 ***
8 REM *** ｹﾞｷﾂｲ ｹﾞｰﾑ ***
10 CLS
20 DIM M$(3):M$(0)=" ":M$(1)="0":M$(2)
="*":M$(3)="▦"
30 TA=20:P$="  0*  000 **** **0  ** 00
****  0 "
40 GOSUB 2000
50 DEF FNA=X+Y*40+15362
100 X=20:GOTO160
110 A$=INKEY$:IFA$=" "THEN200
120 IF A$=CHR$(28)THENX=X+1
130 IF A$=CHR$(29)THENX=X-1
140 IF X<2THENX=2
150 IF X>33 THENX=33
160 CURSOR X-1,22:PRINT" ↑ ";
170 CURSOR X-2,23:PRINT" ▲ ";
180 GOSUB1000:GOTO110
200 Y=22:C=32:TA=TA-1:IF TA<0THEN500
210 CURSOR 25,2:PRINT"TAMA    ":CURSOR2
9,2:PRINTTA
220 VPOKE FNA,32
```

```
230 IF C<>32 THEN 270
240 Y=Y-1:C=VPEEK(FNA):VPOKE FNA,142
250 IF Y<8THEN 400
260 GOSUB1000:GOTO 220
270 BEEP:S=-20
280 IF C=79THEN S=X
290 IF C=42 THEN S=X*2
300 SC=SC+S
310 GOSUB 2000:GOTO160
400 SOUND 5,2,15
410 FOR T=1 TO  50:NEXT T
420 SOUND0:VPOKE FNA,32
430 SC=SC-3*X:GOSUB 2000
440 GOTO 160
500 CURSOR 15,15:PRINT"GAME OVER"
510 GOTO 510
1000 P$=RIGHT$(P$,31):P$=P$+M$(RND(1)*
4)
1010 CURSOR 2,10:PRINTP$
1020 RETURN
2000 CURSOR 10,2:PRINT"SCORE      "
2010 CURSOR 16,2:PRINTSC
2020 RETURN
```

それ以外の部分もいくつかのブロックに分けることができます。

行番号　10～　50	初期設定
行番号100～180	砲台の移動
行番号200～260	弾の発射
行番号270～310	命中したときの処理
行番号400～440	はずれたときの処理
行番号500～510	ゲーム終了

得点の表示は、得点が変更されるごとに行ない、標的の表示は、砲台の移動と弾の発射の両方で行なっています。残りの弾数は、発射するごとに1つずつ減らし、表示します(行番号200と210)。

得点の計算は、なかなかこっています。まず、命中した場合には、命中したキャラクタが“O”（コード79）だったら砲台の位置Xの値がそのまま得点になります。“※”だったらXの値の2倍が得点になります。

残りの“☻”に当たった場合、これはXの値に関係なく、一律、マイナス20点となります。計算方法は、増減の値をSに入れておいてから行番号300で得点の変数SCに加えるというやり方です。

残念ながら弾がはずれてしまった場合は、行番号430で得点からマイナスします。マイナスする点数は、なんとXの値の3倍。手きびしい点にします。

さて、行番号50をみてください。

```
50　　DEF　FNA=X+Y*40+15362
```

ここではユーザー関数FNAを定義しています。式の中にあるXとYは、プログラム中のX（砲台の位置）とY（弾の位置）に一致します。ここでX+Y*40+15362とFNAを定義しておくと、プログラム中でX+Y*40+15362と書かなければならないところが、FNAだけですみます。何度も使うときにはとても便利です。プログラムも短くなり、時間もかからないというわけです。

もう1つちょっと気のきくテクニックを使っています。それは弾の数の表示と、得点の表示の部分です。“SCORE”と実際の得点（行番号2000）、“TAMA”と実際の残った弾の数（行番号210）をそれぞれ別々に表示しています。

PRINT“SCORE”；SCと書けばすむところをわざわざ分けて書いているのには、実はわけがあります。もしPRINT“SCORE”；SCと書いてしまうとちょっとまずいことが起こるのです。それは表示する数字のケタ数の問題です。

得点の場合を例に考えてみましょう。得点のケタ数は1～3ケタまでで、

マイナスの数になりマイナス符号がつくこともあります。このように、そのときどきによって異なるケタ数の数字を同一の場所に表示しようとすると、前の数字が一部残ってしまうことがあるのです。これは数字は左づめで表示されるということから起こる不都合です。

たとえば、初めに20と表示してあったとします。この得点が減って、4になったらどうでしょう。4は1ケタですから20の2のところが4に変わり、0(ゼロ)が残ってしまいます。

そこでまず"SCORE 　　　"として、得点の表示の部分をスペースで消しておき、その後でスペースの部分に得点を表示しているのです。

弾の数の表示も同じように"TAMA"の後にスペースを入れて"TAMA○○○"と表示し、その後、実際の数を表示しています。

このようなことは、初めから考えに入れてプログラムをつくれるものではありません。一度やってみて、ここのところはどうも変だな、という点がでてきたら、その原因をさがして直していくという手順をとるようにしてください。

いよいよ「ホウダイ・ゲーム」のメインともいえる「流れる標的」のつくり方にはいります。

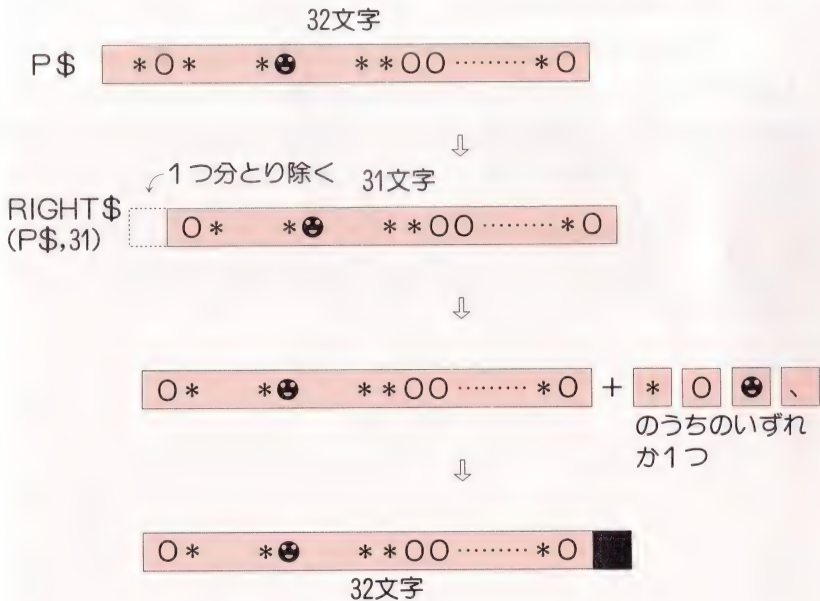
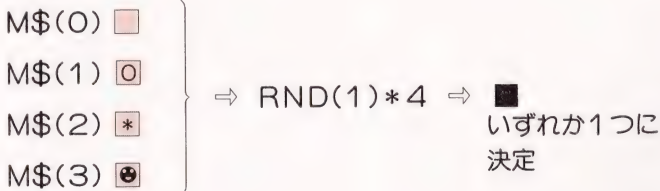
標的の表示は、行番号1000からの部分で行なっています。

```
1000  P$=RIGHT$(P$,31):P$=P$+M$(RND(1)*4)
1010  CURSOR 2,10:PRINT P$
1020  RETURN
```

これを1回実行するごとに、標的を1つ左へずらし、右端に新しい標的を1つ加えて表示します。

P$には、現在の標的のパターンが入っていて、長さは32文字分あります。

行番号1000では、まずRIGHT$(ライトダラー)関数で、左端の1文字を除いた31文字分の標的をとりだしP$に入れます。次に右端に、4つのキャラクタ("O(オー)"、

"※"、"☻"、スペース）のうちのいずれか１つを乱数を使って選びだし、つけ加えます。これで新しい32文字分の長さの標的がP$に入ります。M$は文字の配列変数です。行番号20の初期設定のときに、M$(0)、M$(1)、M$(2)、M$(3)のそれぞれに標的のキャラクタを入れておきます。こうしておくと、つけ加えるキャラクタを決めるときに、0～3の乱数をつくり、配列の添字にし、そのときどきによって異なるキャラクタをつけ加えることができるわけです。

ＲＮＤ関数によってつくりだす数は整数とは限りませんが、添字として使うと、数値は自動的に整数に変換されますので、ＩＮＴ関数を使ってわざわざ整数に直す必要はありません。

さて、標的は右から左へ流れるようにしてありますが、これを逆に左から右へと流れるようにすることもできます。その場合には、標的のパターンの左から31文字をとって左端に１つ標的をつけ加えればいいのです。

```
P$ =LEFT$(P$,31)
P$ =M$(RND(1)*4)+P$
```

M$をP$につなげるときのM$とP$の位置に注意してください。文字列変数を＋でつなぐときには、＋の左と右の順序がそのままつながる順序になります。

M$+P$ ――→ | M$ | P$ |

P$+M$ ――→ | P$ | M$ |

「撃墜ゲーム」いかがでした？　これでバッチリ楽しめますよ。

6 ガッツでCG!

コンピュータ・グラフィック（CG）というと、むずかしい技術が必要のような気がしますが、そんなことはありません。複雑で美しい絵を描こうと思ったら、とにかくガッツです。〝ガッツでCG〟ですね。つまり根気よくデータを打ちこみさえすれば、思いのままにCGの世界が楽しめます。次のプログラムを実行してみてください。

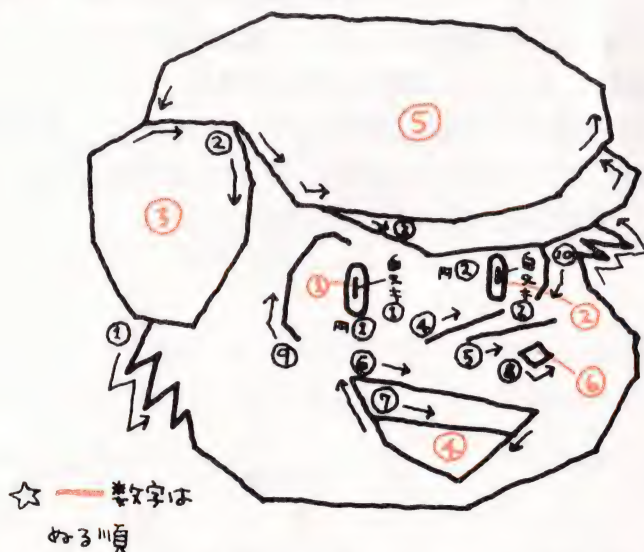
＊8・7　子グマの顔を描くプログラム

```
100 REM *** 8.7 ***
110 REM * ｲﾇ ｶﾞﾎﾞｳｼ ｦ ｶﾌﾞｯﾀ ｴ ｦｶｺｳ *
120 SCREEN 2,2:CLS
130 READ C,B,F:COLOR C,B,,F:CO=C
140 READ A
150 ON ABS(A) GOTO 200,300,400,500,600
,160
160 GOTO 160
200 REM ***LINE
210 READ X,Y:PSET (X,Y),CO
220 READ X:IF X<0 THEN A=X:GOTO 150
230 READ Y
240 LINE-(X,Y),CO
250 GOTO 220
260 RIRCLE (X,Y),R,CO,RA,ST,EN
270 REM ***CIRCLE***
280 CX=R:CY=0:F=INT(-R/2)
300 REM***COLOR
310 READ CO:GOTO 140
400 REM***CIRCLE
410 READ X,Y,R,RA,ST,EN
420 CX=R:CY=0:F=INT(-R/2)
425 IF CY>CX THEN RETURN
430 PSET (X+CX,Y+CY),CO
435 PSET (X-CX,Y+CY),CO
```

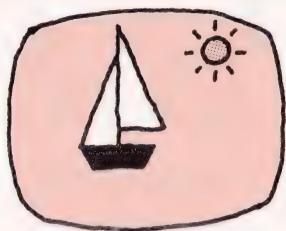
```
440 PSET (X+CX,Y-CY),CO
445 PSET (X-CX,Y-CY),CO
450 PSET (X+CY,Y+CX),CO
455 PSET (X-CY,Y+CX),CO
460 PSET (X+CY,Y-CX),CO
465 PSET (X-CY,Y-CX),CO
470 CY=CY+1:F=F+CY
475 IF F>0 THEN F=F-CX:CX=CX-1
480 GOTO 140
500 REM***PAINT
510 READ X,Y,C
520 PAINT(X,Y),C
530 GOTO 140
600 REM***BLINE
610 READ X,Y:PRESET (X,Y)
620 READ X:IF X<0 THEN A=X:GOTO 150
630 READ Y
640 BLINE -(X,Y)
650 GOTO 620
999 REM***DATA
1000 DATA 1,15,5
1010 DATA -1,28,92,20,106,30,100,25,11
6,34,108,32,118,36,115,40,128,55,140,1
10
1015 DATA 144,147,129
1020 DATA 157,110,158,100,145,85,143,7
6,150,77,145,73,157,75,146,69
1030 DATA -1,52,38,55,54,52,73,39,94,2
5,91,12,70,12,45,26,38,52,38,66,61,108
,66
1035 DATA 138,58,153,44,151,31,127,15,
62,9
1040 DATA 33,23,26,38
1050 DATA -2,12,-1,70,63,103,75,135,73
,155,64,160,55,150,47
1060 DATA -2,9,-1,101,99,123,91
1070 DATA -2,9,-1,113,98,136,93
1080 DATA -1,80,108,90,127,111,137,131
,117,94,108,80,108
1090 DATA -1,85,118,124,124
1100 DATA -1,125,102,130,107,135,102,1
31,100,125,102
```

```
1110 DATA -1,68,97,64,94,65,80,72,70,7
8,69,81,72
1120 DATA -1,132,73,134,76,134,83,132,
86,130,87
1130 DATA -3,82.5,85,2.5,2.8,0,1
1140 DATA -3,121.5,82,2.5,2.8,0,1
1150 DATA -4,82,88,1
1160 DATA -4,121,82,1
1170 DATA -5,83,81,83,85,82,82,82,86
1180 DATA -5,122,78,122,82,122,82,121,
79,121,83
1190 DATA -4,30,60,6
1200 DATA -4,110,130,8
1210 DATA -4,90,40,2
1220 DATA -4,130,103,4,-6
```

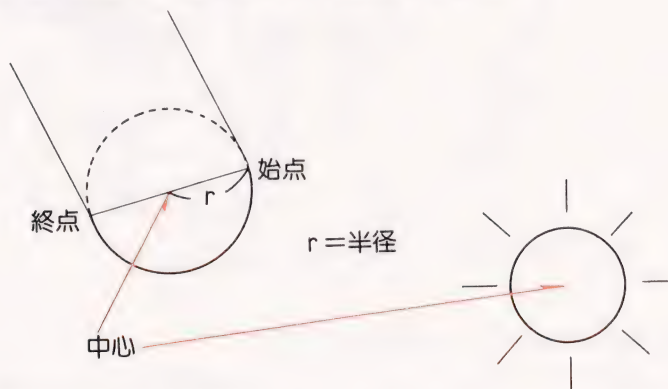
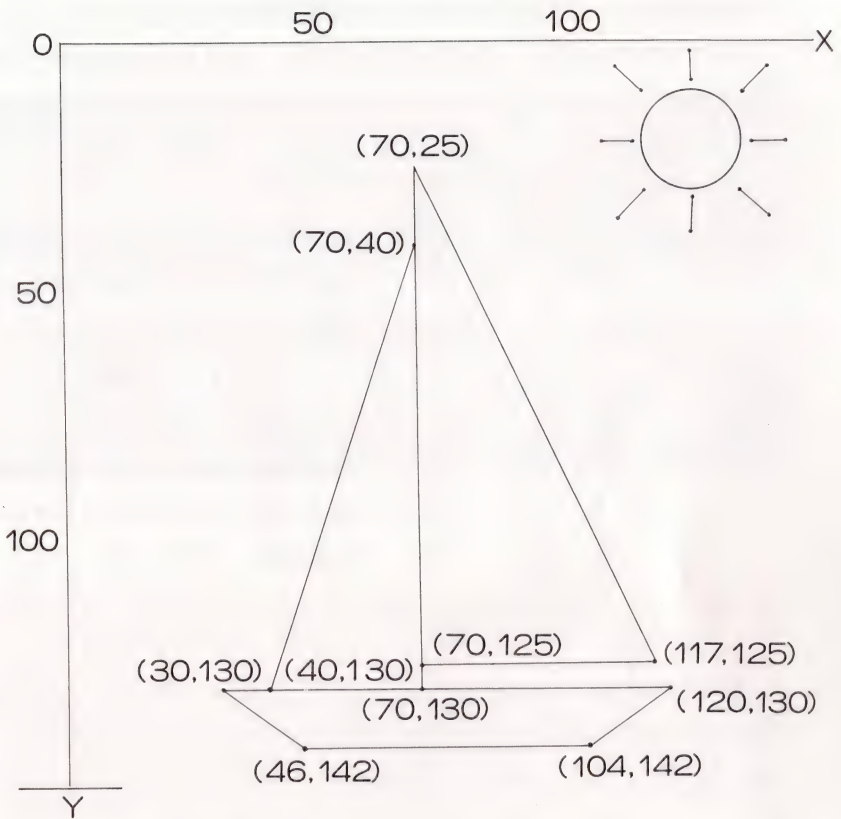
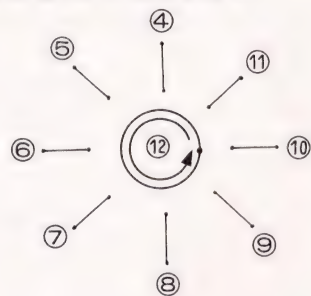
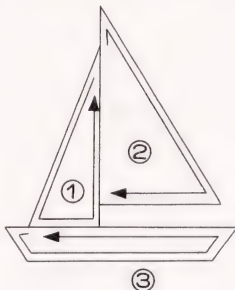
まず画面に子グマの顔が現われますね。

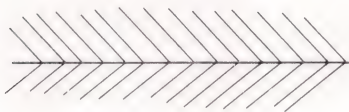
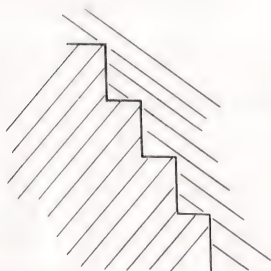
この絵を描こうと思ったら、まず次のように方眼紙に絵を描いて線を引く順序を考えます。

後は、絵をもとにして、線を引いたり色をぬったり、とプログラミングしていきます。“どこからどこまで”とか“どの部分に”といった指定はプログラムの最後のDATA文で次々に与えていきます。とにかく、このDATA文で与えるたくさんのデータがCGのミソです。

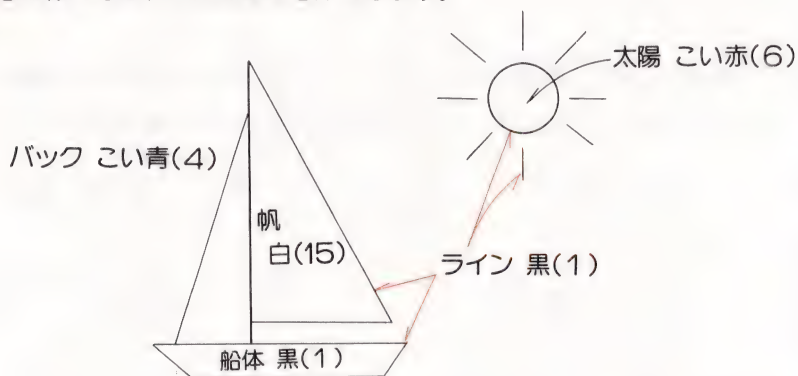
それでは、もう少し単純なプログラムを例にして、データの入力方法を説明していきます。ここでは、ヨットに太陽、といきましょう。

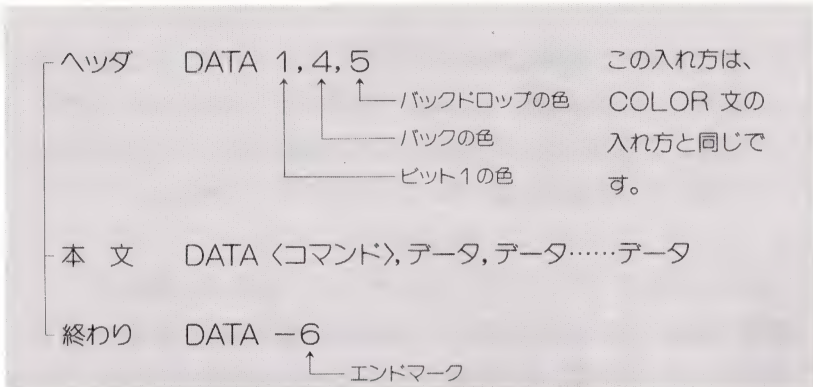
① 大まかな原画を、25.6cm×19.2cmの方眼紙の上に描きます。

② 座標を決めて、正確に描きます。

・直線を使って描きます。点と点を、定規で直線を引きながら、むすんでいきましょう。なめらかな線を描きたいときには、ちょっと面倒ですが、点と点の間隔を小さくして、直線でむすびます。
直線で、ヨットと、太陽の光が描けますね。

・円や円弧を描くことができますので、円弧を使って、曲線も描けます。ただし、これらを使うときには、円（または円弧）の中心、半径、始点、終点の4つの値を求めなければなりません。

・全部の折れ線の座標を求めます。左上隅を原点とし、1㎜＝1ドットと考えましょう。

・なるべく長く一筆書(ひとふでが)きできる部分をさがして、全体を何本かの線に分けてください。そして、分けた線の始まりと終わりを決めます。

③　次は、ぬりつぶす部分と各ラインの色を決めましょう。

ぬりつぶすときには、ヨコ方向8ドットにつき1色しか使えませんので、注意してください。たとえば、となり合う部分を、両方ぬりつぶしたとします。

水平な線や傾きがゆるやかな線の場合には、きれいにぬり分けられます。

傾きが急な線は、さかい目が階段のように、ガタガタになってしまいます。

傾きが急な線は、うまくぬり分けられませんね。けれども、となり合う部分を、両方ぬりつぶすのではなく、片方だけぬりつぶすとしたら、急な線でもきれいにぬることができます。

(　)内は色番号

④　①～③まで、すべて決まったら、DATA 文として、さっきのプログラムに入れてみましょう。DATA 文の入れ方は、次のとおりです。

まずは、次のプログラムを見てください。

＊8・8　DATA 文の入ったプログラム

```
1000 REM *** 8.8 ***
1005 REM *** ヨット ヲ カイテミヨウ ***
1010 DATA 1,4,5
1020 DATA -1,70,40,40,130,70,130,70,40
1030 DATA -1,70,40,70,25,117,125,70,125
1040 DATA -1,40,130,30,130,46,142,104,
142,120,130,70,130
1050 DATA -4,60,100,15
1060 DATA -4,80,100,15
1070 DATA -4,75,135,1
1080 DATA -2,1,-1,125,2,125,8
1090 DATA -1,110,5,115,10
1100 DATA -1,107,20,113,20
1110 DATA -1,110,35,115,30
1120 DATA -1,125,32,125,38
1130 DATA -1,135,30,140,35
1140 DATA -1,137,20,143,20
1150 DATA -1,140,5,135,10
1160 DATA -3,125,20,10,1,0,1
1170 DATA -4,125,20,6
1180 DATA -5,70,40,70,125,-6
```

これは、ヨットと太陽を表示するプログラムのDATA文の部分です。DATAの次に、必ず−1とか−4などの数字が出てきますね。これが、コマンドです。このコマンドが、実はたいへん重要なのです。次の説明を、じっくり読んで、あなた自身のデータを入れてみてください。

−1 連続した線を引きます。−1の後に、②で決めた座標X、Yを、入れていきます。たとえば、ヨットの左の帆を描くのには、次のように入力します。

DATA　−1, 70, 40, 40, 130, 70, 130, 70, 40

↑区切りは、カンマで。

データ数は、どんなに長くなってもかまいませんが、必ず、X、Y1組ずつ並べて入れなければなりません。

−2 ラインの色を指定します。このコマンドを使って色を指定すると、それ以降に指定するラインや円弧は、全部その色になります。途中で、また色指定をすると、今度はその色でラインが描けます。

たとえば、太陽の光を黒（色番号1）で描きたいときには、次のように入力します。

DATA　−2, 1, −1, 125, 2, 125, 8

↑ポイントはここ！

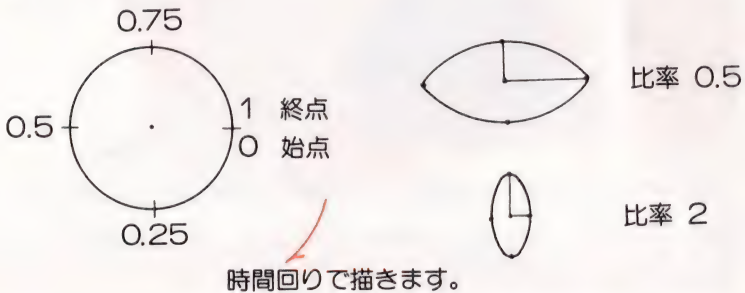
この指定の後は、次の色指定まで、ずっと黒で線（ライン）を描いていきます。

－3 円、円弧を描きます。

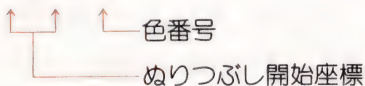
たとえば、太陽を描くときには、次のように入力します。

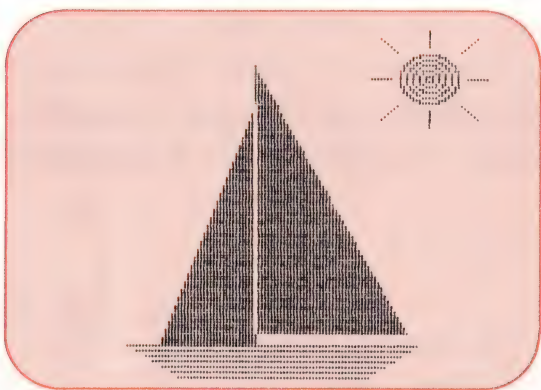
DATA　−3, 125, 20, 10, 1, 0, 1

中心と半径の出し方は、もう知っていますね。終点と始点というのは、円または円弧の描き始めの位置と、描き終わりの位置です。

また、比率というのは、ヨコの長さに対するタテの長さの割合です。

－4 ぬりつぶしの部分と、色を指定します。

たとえば、ヨットの左の帆を、白くぬりつぶしたいときには、次のように入力します。

DATA　−4, 60, 100, 15

ぬりつぶしの指定は、－2の色指定とは別です。ぬりつぶし指定をした後、あらためて色指定をし直す必要はありません。

－5 1度描いた部分を消します。1度ぬりつぶした部分は、他の色で線を引くことはできません。そこで、ぬりつぶした部分に、何かを描くときには、このコマンドを使って、バックの色で描きます。DATA文の入れ方は、－1のときと同じです。

もうすっかり
夏の気分！

どうですか？ちょっと面倒くさいけれど、むずかしくはないでしょう。最初に示した、メインプログラムを活用して、あなた自身のコンピュータ・グラフィックの世界をひろげてみてください。

9

スプライト機能でパワーアップ！

本格派ゲームの切り札！　おもしろさ倍増だよ!!

スイッチオン！　する前に

ゲームやアニメーションづくりに大きな手助けをしてくれるのが、スプライト機能(きのう)です。だれですか？　清涼飲料水じゃないの？　なんて言ってるのは。実はスプライト機能があると、グラフィック画面上をキャラクタが自由に動いたり、すれちがったりすることができる大変便利なものです。きっと、プログラムづくりがますますおもしろくなって、ゲームやアニメーションもパワーアップしますヨ！

1 スプライトはこうして使う

SK-1100は、ベーシックSKⅢというベーシックが使えるんです。このベーシックSKⅢではスプライト機能というものを使うことができますが、ここではそれをご紹介しましょう。スプライト機能を使うと、ゲームやアニメーションが、よりいっそうおもしろいものになります。

スプライト機能というのは、グラフィック画面上にキャラクタをつくり、そのキャラクタを動かす働きです。ここで注目したいのは、キャラクタは画面に描いた背景の上に表示しても、下の絵と重なり合ってしまうことがないということです。背景の上をちゃんと動いてくれますので、ゲームやアニメーションが本格的なものになります。

●キャラクタのつくり方

まず、キャラクタのつくり方をみてみましょう。これは、これまでにキャラクタをつくったのと同じように、PATTERN(パターン)文を使います。

PATTERN　S#0, "030F1F3F7E7EFFFF"

S#は、スプライト機能を示しています。次の0がつくろうとしているキャラクタの番号で、スプライト名称といいます。キャラクタは、全部で256個つくることができます。そこでキャラクタ番号は、256個分、つまり0～255までの数字で指定します。

スプライト名称の後の" "内は、キャラクタのパターンを示しています。8×8ドットの枠内にパターンを描く方法は、すでにおなじみですね。

さて、このスプライト機能でキャラクタをつくるときには、4種類の大きさを使いわけることができます。この使いわけをするのが、MAG(マグ)文です。

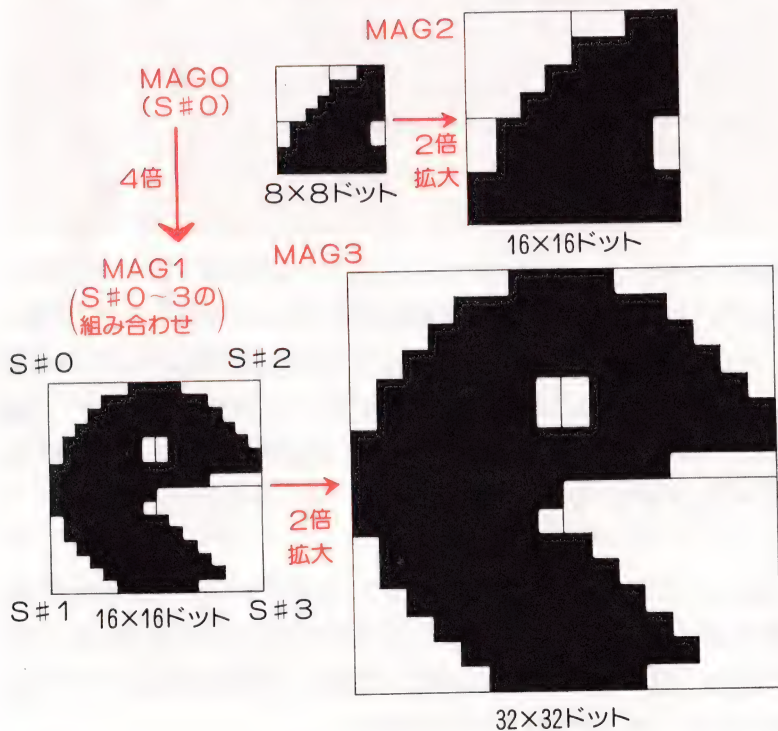
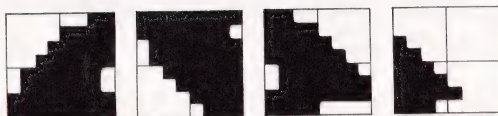
次のプログラムを実行してみてください。

＊９・１　スプライト機能を使ってキャラクタをつくってみよう

```
10 REM *** 9.1 ***
20 REM *** ﾊﾟｯｸﾏﾝ ｦ ﾂｸｯﾃﾐﾖｳ ***
30 SCREEN2,2:CLS
40 PATTERN S#0,"030F1F3F7E7EFFFF"
50 PATTERN S#1,"FEFF7F7F3F1F0F07"
60 PATTERN S#2,"C0F0F8FC7E7FFFF0"
70 PATTERN S#3,"000080C0E0F0F8E0"
80 FOR M=0 TO 3
90 CLS:MAG M
100 CURSOR 100, 30:PRINT"MAG";M
110 SPRITE 0,(100,100),0,1
120 FOR T=1 TO 500:NEXT T
130 NEXT M
140 GOTO 80
```

画面の中央近くに、４種類のキャラクタがでてきます。キャラクタの大きさは、このMAG0～MAG3の４種類があります。このキャラクタを見ると、MAG0とMAG2、MAG1とMAG3の絵が同じですね。MAG0を2倍に拡大したものがMAG2、MAG1を2倍に拡大したものがMAG3という関係になっています。また、MAG1はMAG0の４倍の大きさで、これはMAG0の大きさのキャラクタを４つ組み合わせてつくったものです。

つまり、PATTERN文でつくったキャラクタのS＃0をそのまま表示したものがMAG0、それを2倍に拡大したものがMAG2になっています。また、S＃0～S＃3の４つを組み合わせて表示したものがMAG1、それを2倍に拡大したものがMAG3になります。

PATTERNで
つくったキャラクターを
くわしく見ると……
MAG2
MAG0
(S#0)
2倍
拡大
4倍
8×8ドット
16×16ドット
MAG3
MAG1
(S#0~3の
組み合わせ)
S#0
S#2
2倍
拡大
S#1
16×16ドット
S#3
32×32ドット

MAG文のつくり方は、このようにMAG0（ゼロ）～3というように指定します。MAG0～3をまとめておきましょう。

MAG0……1ドットを1ドットとして、8×8ドットの枠に絵を描く。

MAG1……8×8ドットのパターンを4つ組み合わせて、16×16ドットの枠に絵を描く。組み合わせるパターンはS＃0～3、S＃4～7のように続く番号のパターンを組み合わせることになります。

MAG2……2×2ドットを1ドットとして、16×16ドットの枠（わく）に絵を描く。

MAG3……MAG2で指定した枠の絵を4つ組み合わせて、32×32ドットの枠に絵を描く。

●キャラクタを動かしてみよう！

スプライトを使ってキャラクタをつくる方法がわかったところで、次はこのキャラクタを画面に表示して動かす方法をみてみましょう。

キャラクタの表示や移動を行なうのは、SPRITE（スプライト）文になっています。

```
SPRITE    0, (100, 104), 0, 1
```


スプライト面番号については、この後くわしく説明しますので、ここではちょっとおいておいて、次の座標(100、104)をみてください。これは、グラフィック画面の座標で、X座標とY座標で示します。その次のスプライトパターンナンバーは、PATTERN文のS＃で指定した番号（スプライト名称）です。最後の表示カラーは、文字どおり表示するときの色で、カラーコードで指定します。

MAG1やMAG3を使って4つのパターンを組み合わせて表示しようという場合には、MAG文でまず、MAG1またはMAG3と指定しておきます。そして、次にSPRITE文で、4つのパターンの中の一番小さいスプライトパターンナンバーを指定します。たとえば、ナンバーに8と指定すると、S#8、S#9、S#10、S#11の4つのパターンを組み合わせて表示します。ナンバーは、4つがひと組になりますので、指定するときには、0、4、8、12……と4の倍数(ばいすう)で指定することになります。

スプライト機能を使ったことにより、これまでと画期的にちがう点があります。

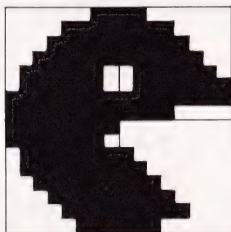
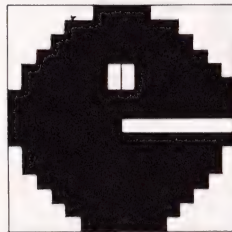
キャラクタを動かすときに、前に描いたキャラクタを消してから移動する位置に新たにキャラクタを描くというのがこれまでのやり方でした。ところが、スプライト機能を使うと、前に描いたキャラクタを消す必要がありません。移動する位置の座標を次々に指定するだけで、キャラクタが移動します。便利でしょう?

じゃ、次のプログラムをみてください。

＊9・2　8つのキャラクタをつくると

```
10 REM *** 9.2 ***
20 REM *** ﾊﾟｯｸﾏﾝ ｶﾞ ｸﾁｦ ﾊﾟｸﾊﾟｸ ***
30 SCREEN2,2:CLS
40 MAG1
50 PATTERN S#0,"030F1F3F7E7EFFFF"
60 PATTERN S#1,"FEFF7F7F3F1F0F07"
70 PATTERN S#2,"C0F0F8FC7E7FFFF0"
80 PATTERN S#3,"000080C0E0F0F8E0"
90 PATTERN S#4,"030F1F3F7E7EFFFF"
100 PATTERN S#5,"FFFF7F7F3F1F0F07"
110 PATTERN S#6,"C0F0F8FC7E7EFFFF"
120 PATTERN S#7,"00FFFEFEFCF8F0E0"
```

```
130 FOR X=0 TO 255 STEP16
140 FOR I=0 TO 7
150 SPRITE 0,(X+I,100),0,1
160 NEXT I
170 FOR I=8 TO 15
180 SPRITE 0,(X+I,100),4,1
190 NEXT I
200 NEXT X
210 GOTO 130
```

ここでは、8つのキャラクタをつくっています。MAG1を指定していますので、S＃0～3、S＃4～7と4つをひと組にして16×16ドットのキャラクタを2つ表示する形になります。

S＃0（ゼロ）－S＃3

S＃4－S＃7

行番号150ではS＃0（ゼロ）～3のキャラクタを、行番号180ではS＃4～7のキャラクタを表示し動かしているのは、もうおわかりですね。

表示する位置の座標は、XとIを使って変化させ、それによってキャラクタが少しずつ動いていきます。

2つのキャラクタは、8ドットの間隔をあけたままで1ドットずつ画面の左から右へ（X座標の0から255へ）動きます。このときそれぞれのキャラクタは8ドット進むと、8ドット飛びこし、また少しずつ動いていきます。

●2つのキャラクタの動き

Xの値	0		16	
Iの値	01234567	89101112131415	01234567	891011…………
X座標 (X+Iの値)	01234567	89101112131415	1617181920212223	242526……………

●こんなに楽しいアニメがつくれるよ！

それでは、MAG3の大きなキャラクタを使ったプログラムを2つご紹介します。どちらも楽しいアニメーションです。

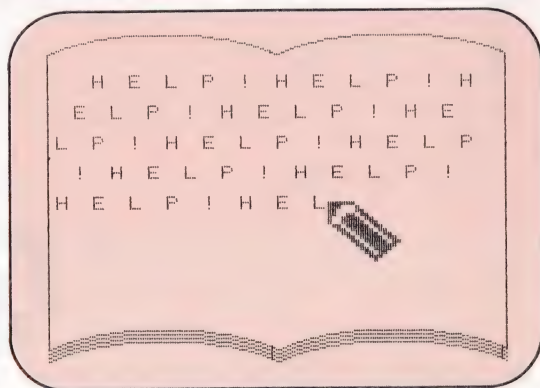
さあ、実行してみましょう。

＊9・3　ノートに文字を書くえんぴつが描けるよ

```
5000 REM *** 9.3 ***
5010 REM * ｴﾝﾋﾟﾂ ﾃﾞ ﾉｰﾄﾆ ｼﾞｦ ｶｲﾃﾐﾖｳ *
5010 SCREEN 2,2:CLS
5020 COLOR 15,1,(0,0)-(255,191),1
5030 GOSUB 5280
5040 X=16:Y=24:E=36
5050 PATTERNS# 0,"FCE2C3DD9E9F6F37"
5060 PATTERNS# 1,"1B0D060301000000"
5070 PATTERNS# 2,"00000080C060B0D8"
5080 PATTERNS# 3,"ECF6FB7EBCD87020"
5090 MAG 3:N=1
5100 READ A$(1),A$(2),A$(3),A$(4),A$(5
)
5110 SPRITE 1,(X,Y),0,15
5120 X=X+2:Y=Y+1
```

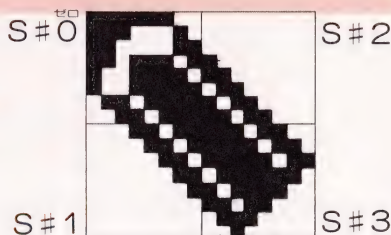
```
5130 IF Y>=180 THEN 5010
5140 IF Y=E THEN Y=Y-8:CURSORX,Y:PRINT
 A$(N):N=N+1:IF N>=6 THEN N=1:RESTORE
5150 IF X>=200 THEN X=16:GOSUB 5170
5160 GOTO 5110
5170 Y=Y+16:E=E+16
5180 RETURN
5190 DATA H,E,L,P,!
5280 SCREEN 2,2:CLS
5290 X=0:Y=0
5300 CIRCLE(70,30),60,15,.4,.6,.9
5310 CIRCLE(170,30),60,15,.4,.6,.9
5320 FOR N=0 TO 6 STEP 2
5330 CIRCLE(70,185+N),60,15,.4,.6,.9
5340 CIRCLE(170,185+N),60,15,.4,.6,.9
5350 NEXT N
5360 LINE(20,17)-(20,171),15
5370 LINE(219,17)-(219,171),15
5380 RETURN
```

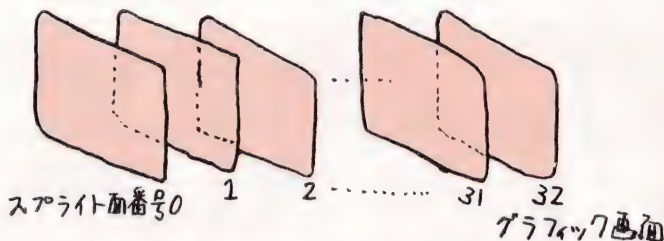
このプログラムを実行すると、ノートの中にスプライト機能でつくったえんぴつのキャラクタがあらわれて、“HELP！HELP！HELP！……”と書きはじめます。えんぴつが“助けて！”なんておかしいですね。

I am a pencil.
Help me！

ノートに文を書いたえんぴつのキャラクタをつくって、今度はすきな文字を画面に書いてみます。それでは、実行しましょう。RUN！

＊9・4　キミのすきな文字が書けるよ

```
5000 REM *** 9.4 ***
5010 REM *** キミ ノ スキナ ジ ヲ カイテミヨウ ***
5020 SCREEN 2,2:CLS
5030 COLOR 15,1,(0,0)-(255,191),1
5040 X=24:Y=0:E=8:I=16
5050 PATTERNS# 0,"FCE2C3DD9E9F6F37"
5060 PATTERNS# 1,"1B0D060301000000"
5070 PATTERNS# 2,"00000080C060B0D8"
5080 PATTERNS# 3,"ECF6FB7EBCD87020"
5090 MAG 3:N=1
5100 SPRITE 1,(X,Y),0,15
5110 CURSOR I,L:PRINTA$:GOSUB 5190
5120 SPRITE 1,(X,Y),0,15
5130 X=X+1:Y=Y+1
5140 IF Y>=180 THEN 5020
5150 IF Y=E THEN Y=Y-8:GOTO 5110
5160 IF X>=240 THEN X=24:GOTO  5180
5170 GOTO 5120
5180 Y=Y+16:E=E+16
5190 REM
5200 A$=INKEY$
5210 IF A$<>"" THEN 5230
5220 GOTO 5200
5260 FOR W=0 TO 10 :NEXT W
5270 I=I+8:IF I>=248 THEN I=16:L=L+16:
GOTO 5100
5280 RETURN
```

えんぴつのキャラクタをのせておきます。

2 優先順位でこっち側とあっち側

ここで、SPRITE文を使うときに最初に指定したスプライト面番号について少しお話ししましょう。

スプライト面は32枚あり、この32枚がグラフィック画面上に重なっています。SPRITE文でキャラクタをつくるときは、スプライト面1枚の上に描く仕組みになっています。

さて、スプライト面の1と2にキャラクタを描いたとします。この2つのキャラクタを動かしたときに、キャラクタ同士が交差(こうさ)すると、後ろのスプライト面のキャラクタ、つまり2の面に描いたキャラクタがかくれてしまいます。

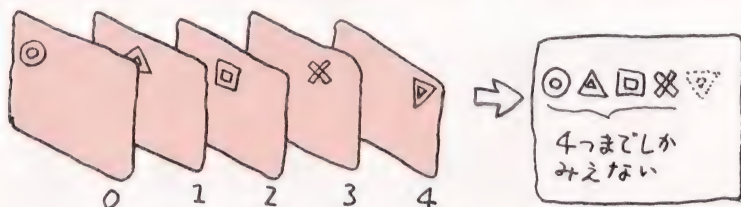
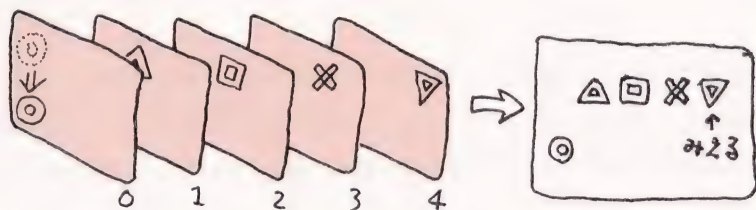
このようにスプライト面のキャラクタが交差した場合には、スプライト面番号の小さい方のキャラクタが前になります。どちらが前にくるのかということを優先順位(ゆうせんじゅんい)といいます。「優先順位が高い」というと、交差したときに前にきます。

つまりスプライト面は番号の小さいほうが優先順位が高いということになります。

32枚のスプライト面に同時にキャラクタを表示することができますので、

いっぺんに32個ものキャラクタが画面にあらわれる、なんていうこともできるわけです。

ただし、ここで１つだけ制約があります。

それは、"同一水平線上には４つのキャラクタしか表示できない"ということです。もし４つ以上のキャラクタが並ぶと、優先順位の高いスプライト面のキャラクタ４つを表示して、あとは消えてしまいます。

このときスプライト面０のキャラクタを下にずらしてみます。そうすると水平線上には、１～４のキャラクタの４つになりますので、スプライト面４のキャラクタも表示します。

ゲームやアニメーションをつくるときに、このことは制約にもなりますが、逆にこのことを使っておもしろ味をだすこともできます。

つまり、スプライト面０のキャラクタを動かすことによって、スプライト面４のキャラクタを出したり消したりすることができます。これをゲームやアニメーションに活かしたらどうかな、ということなのです。

まだまだ、この優先順位は工夫のしかたしだいでいろいろ楽しい使い方ができます。ちょっと頭をひねってみてくださいね。

3 4色キャラクタが動く

スプライト機能を使ってつくったキャラクタをおもしろく使うこともできます。それが4色キャラクタです。4つのキャラクタを組み合わせて使えるという性質を活かして、4つの色を使ったキャラクタをつくってみようというわけです。

まず、4つの色を重ねたときに、どんな具合になるのかみてみましょう。次のプログラムを実行してください。

＊9・5　4つの色を重ねると……

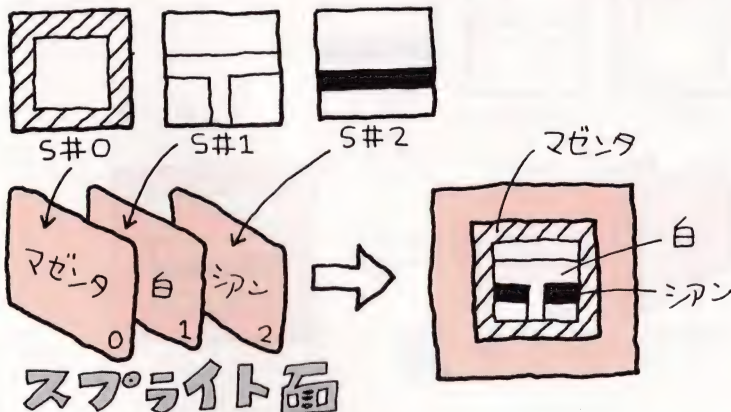
```
10 REM *** 9.5 ***
20 REM *** 4ツ ノ イロ ガ゛ ヌレルヨ ***
30 SCREEN2,2:CLS
40 MAG2
50 PATTERN S#0,"6060606060606060"
60 PATTERN S#1,"00FFFF0000000000"
70 PATTERN S#2,"0606060606060606"
80 PATTERN S#3,"0000000000FFFF00"
90 FOR I=0 TO 3
100 SPRITE 0,(100,100),I,I+1
110 GOSUB 180
120 NEXT I
130 FOR I=0 TO 3
140 SPRITE I,(100,100),I,I+1
150 GOSUB 180
160 NEXT I
170 GOTO 170
180 FOR T=1 TO 300
190 NEXT T:RETURN
```

はじめに4つのキャラクタをばらばらに表示し、その後重ねて表示します。色は、S#0～3に対して1～4のカラーコードを指定しています。

さて次は、このような多色刷りのキャラクタを動かしてみます。キャラクタを重ねたままで、いっしょに移動します。

＊9・6　キャラクタを重ねて移動させよう

```
5 REM *** 9.6 ***
10 REM *** ヒダ゛リ カラ ミキ゛ ヘ ウコ゛クヨ ***
20 SCREEN2,2:CLS
30 MAG2
40 FOR I=4 TO 120 STEP4
50 CIRCLE (128,96),I,1
60 NEXT I
100 PATTERN S#0,"FF818181818181FF"
110 PATTERN S#1,"0000FFFF18181818"
120 PATTERN S#2,"00000000FFFF0000"
300 FOR X=0 TO 255
320 SPRITE 0,(X,100),0,13
330 SPRITE 1,(X,100),1,15
340 SPRITE 2,(X,100),2,7
360 NEXT X
370 GOTO 300
```

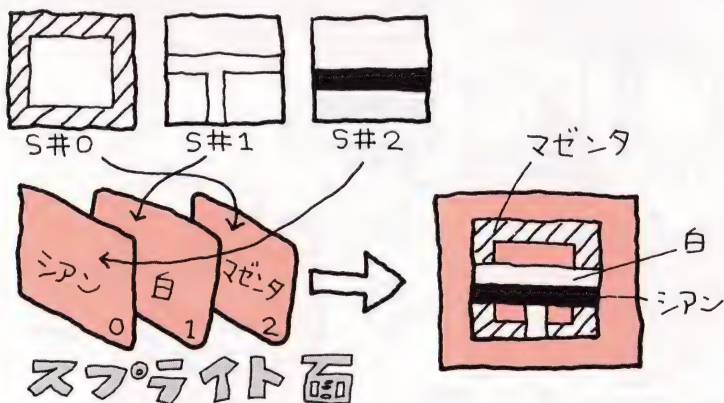
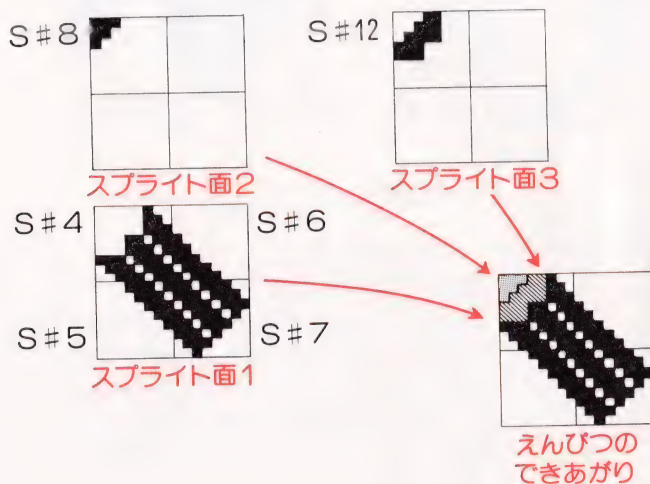
3つのキャラクタを重ねたまま同時に移動していくために、SPRITE文で指定する3つのキャラクタの座標は、3つとも同じにしてあります。

3色キャラクタが左から右へ動いていきますが、このとき背景に描いてある同心円(どうしんえん)が消されないことに注目してください。このようにキャラクタは、背景を消したりすることなく、背景の前を動くことができるのです。

また、キャラクタの中の透明の部分（色をつけていない部分）は、背景がすけて見えます。

キャラクタに自動車の絵を描いたときのことを思い描いてみてください。窓からは、むこう側の風景（背景）がちゃんと見えかくれする、なんていうことができると思いませんか？

ここでキャラクタを書きこむスプライト面を変えてみましょう。同じキャラクタを重ねるにしても、書きこむスプライト面を変えると、またちがったキャラクタができあがります。

スプライト面0(ゼロ)、1、2に、キャラクタのS＃2、1、0を表示します。

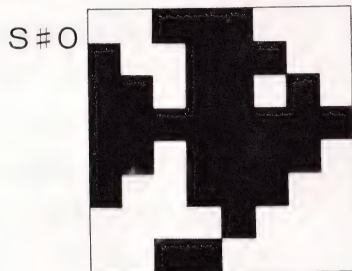
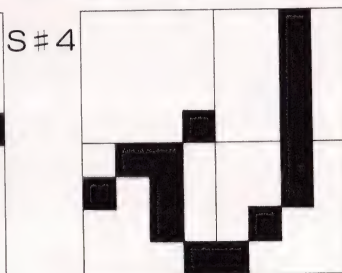
```
320　SPRITE 0、(X、100)、2、7
330　SPRITE 1、(X、100)、1、15
340　SPRITE 2、(X、100)、1、13
```

MAG文の説明のところで実行したえんぴつのプログラムを覚えていますか。ここでもう１度えんぴつのキャラクタの登場です。ただし、そのまままたあのえんぴつを使ったのでは、おもしろくありません。せっかく４色キャラクタや３色キャラクタを覚えたのですから、えんぴつにも色をつけて、赤えんぴつにしてみました。３色にするため、スプライト面は３枚使います。

次のプログラムを実行すると、赤えんぴつが画面の左から右へ、少しずつ移動していきます。この先はおまかせします。

＊9・7　赤えんぴつが左から右へ移動するよ

```
5000 REM *** 9.7 ***
5005 REM *** ｱｶｴﾝﾋﾟﾂ ｶﾞ ｳｺﾞｲﾀ! ***
5010 SCREEN 2,2:CLS
5020 COLOR 15,1,(0,0)-(255,191),15
5030 X=16:Y=80
5040 GOSUB 5120
5050 MAG 3
5060 SPRITE 2,(X,Y),8,6
5070 SPRITE 3,(X,Y),12,10
5080 SPRITE 1,(X,Y),4,9
5090 X=X+1
5100 IF X>255 THEN X=16
5110 GOTO 5060
5120 REM -- PENSHIL ---
5130 PATTERNS#4,"0406071B1DEE773B"
5140 PATTERNS#5,"1D0E070301000000"
5150 PATTERNS#6,"00000080C0E070B8"
5160 PATTERNS#7,"DCEE77BADCE87020"
5170 PATTERNS#8,"E0C0800000000000"
5180 PATTERNS#12,"183878E0E0000000"
5190 RETURN
```

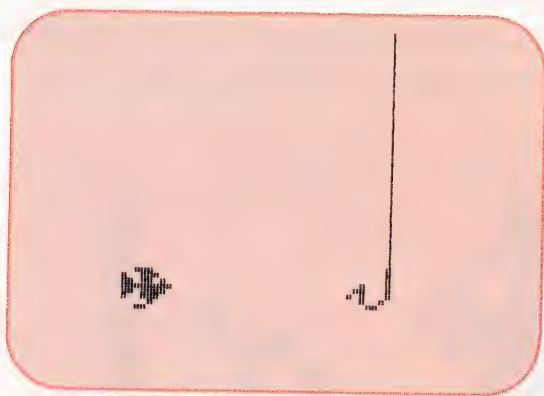
赤えんぴつのアニメーションの次は、“さかなつりゲーム” です。

画面に釣りばりとさかなが登場。これはもちろんスプライト機能を使ったキャラクタです。さかなが釣りばりにひっかかると、画面に“HIT！”の文字がでます。

大きさは、MAG2にして16×16ドットで表示します。

＊9・8　釣りばりを使ってさかなを釣ろう

```
5 REM *** 9.8 ***
10 REM *** ｻｶﾅﾂﾘ ｹﾞｰﾑ ***
20 SCREEN 2,2 :CLS
30 PATTERNS#0,"389CDAFFDE9C0830"
40 PATTERNS#4,"0202022262A22418"
50 X=80:Y=90
60 LINE(163,0)-(163,90),10
70 MAG2
80 SPRITE 0,(X,Y),0,5
90 SPRITE 2,(150,88),4,8
100 X=X+1
110 S=INP(&HBF) AND &H20
120 IF S<>0 THEN GOTO 140
130 GOTO 80
140 CLS
150 CIRCLE(120,90),50,6,.3
160 CURSOR110,85:PRINT"HIT !"
170 FORW=0 TO 100:NEXT W
180 CLS:GOTO 20
```

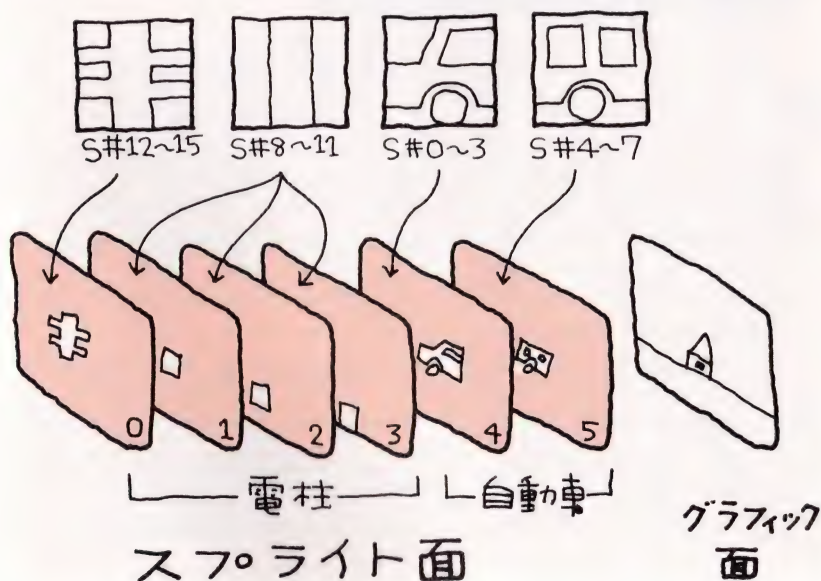
（行番号110、120はスプライトの衝突判定です）

では、RUN!

4 そこのけ、そこのけ、クルマが通る

では、この章のおさらいとして、スプライト機能を利用して、電柱のむこう側を自動車が走り抜けるというアニメーションをつくってみましょう。

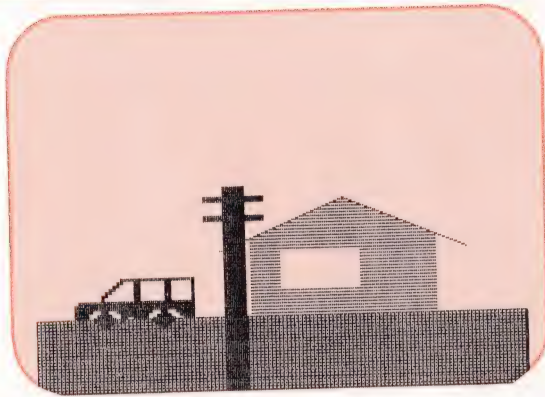
背景となるグラフィック面には、LINE(ライン)文やPAINT(ペイント)文を使って、家と道を描きます。スプライト面には、電柱と自動車を描いて表示します。

自動車は動くようにするのでスプライト面を使うという理由がわかりますが、電柱の方はなぜでしょう。わかった人もいるかな？　自動車が電柱のむこう側を通るようにしたいので、電柱には、自動車を描くスプライト面より優先順位の高いスプライト面を使って自動車の前に立っているようにするのです。

こうすると、自動車は家と電柱の間をスイスイと通りぬけていきます。

＊9・9　クルマが家と電柱の間を通っていくよ

```
100 REM ***.9.9 ***
105 REM *** ｸﾙﾏ ｶﾞ ｽｲｽｲ ﾊｼｯﾃｲｸﾖ ***
110 SCREEN2,2:CLS
120 MAG3
130 GOSUB 1000
160 FOR X=255 TO 32 STEP -1
170 SPRITE 5,(X,120),4,4
180 SPRITE 4,(X-32,120),0,4
190 NEXT X
200 GOTO 160
1000 REM***********ﾊｲｹｲ
1010 LINE(0,144)-(255,144),1
1020 LINE(96,104)-(160,72),1
1030 LINE-(224,104),1
1040 LINE(112,96)-(208,144),11,BF
1050 LINE(128,104)-(168,128),7,BF
1060 LINE(0,145)-(255,191),3,BF
1070 PAINT(160,73),11
2000 PATTERN S#0,"000000000001017F"
2010 PATTERN S#1,"FFFFF9F3F7070301"
2020 PATTERN S#2,"0F183060C08000FF"
2030 PATTERN S#3,"FFFF9FCFEFE0C080"
2040 PATTERN S#4,"FFC0C0C0C0C0FF"
2050 PATTERN S#5,"FFFFF9F3F7070301"
2060 PATTERN S#6,"FFC1C1C1C1C1FF"
2070 PATTERN S#7,"FFFF9FCFE7E0C080"
2080 PATTERN S#8,"0707070707070707"
2090 PATTERN S#9,"0707070707070707"
2100 PATTERN S#10,"E0E0E0E0E0E0E0E0"
2110 PATTERN S#11,"E0E0E0E0E0E0E0E0"
2120 PATTERN S#12,"070707FFFF070707"
2130 PATTERN S#13,"07FFFF0707070707"
2140 PATTERN S#14,"E0E0E0FFFFE0E0E0"
2150 PATTERN S#15,"E0FFFFE0E0E0E0E0"
```

```
3000 REM**********デ゛ンチュウ
3010 SPRITE 0,(88,64),12,14
3020 SPRITE 1,(88,96),8,14
3030 SPRITE 2,(88,128),8,14
3040 SPRITE 3,(88,160),8,14
3050 RETURN
```

ブッブー
オイラ
神風タクシーだあい！

ここでは、電柱は4つ、自動車は2つのスプライト面を使います。

また、MAG3にして、1つのキャラクタを32×32ドットで描き、自動車は、それを2つ組み合わせてつくります。電柱も2つのキャラクタを使っていますが、そのうちの1つを3つのスプライト面に共通に使っています。

自動車を動かすには、前と後ろのキャラクタを32ドット間隔をおいて表示し、同時に1ドットずつ位置を変えて表示していきます。

いかがでしょうか。なかなか楽しいアニメーションになりますね。

キャラクタは256種類中16種類、スプライト面は32枚中6枚を使っただけですので、まだまだ余裕があります。もっとすてきな、美しいアニメーションもつくれます。

さあ、これからはみなさんの腕のみせどころです。スプライト機能をフルに使いこなして、自由にアニメーションの世界を楽しんでください。

(株)システムハウス・オプト

昭和57年2月、登記、設立。マイクロコンピュータのソフトウェアに関する企画・開発、マニュアルやテキストの企画・製作等を手がけている。コンピュータメーカー、出版社、デパート等で精力的に業務を展開中。

SEGA SK-1100による

ゲームとアニメーションのつくり方　定価1200円

昭和59年11月25日　初版発行

著者　(株)システムハウスオプト

発行者　中村　洋一郎

発行所　株式会社　日本実業出版社

東京都文京区本郷3丁目2番12号 〒113
☎ 代表03(814)5161 振替 東京7-25349
大阪市北区西天満6丁目8番1号 〒530
☎ 代表06(362)6141

印刷所　壮光舎印刷株式会社
製本所　株式会社若林製本工場

落丁，乱丁本はお取替え致します

ISBN4-534-00985-2 C0055 ¥1200E

これはおまけ、おこづかい帳！

ゲームやアニメーションとはちょっと違いますが、おまけとして、画面の使い方がおもしろい「おこづかい帳」のプログラムをご紹介しましょう。ひとつ、自分で入力して実行してみてください。

```
10 REM *** ｺｽﾞｶｲ ﾁｮｳ ***
30 CLS:CONSOLE0,24:X=0:Y=6
40 CURSOR7,0:PRINT "ｺｽﾞｶｲ ﾁｮｳ"
50 CURSOR22,0:PRINT "ｵﾜﾘ ﾊ ｼﾅﾓﾉ ﾆ 0"
60 CURSORX,1:PRINT"----------------------------
-----------
70 CURSOR5,2:INPUT"ｺﾝｹﾞﾂ ﾌﾞﾝ ¥";K
80 CURSORX,3:PRINT"----------------------------
-----------
90 CURSOR 3,4:PRINT"ｺｳﾓｸ"
100 CURSOR 18,4:PRINT"ｷﾝｶﾞｸ"
110 CURSOR 30,4:PRINT"ﾉｺﾘ"
120 CONSOLE5,19
130 CURSOR0,21:PRINT"-------------------------
---------------
140 CURSOR0,22:INPUT"ｼﾅﾓﾉ ";S$
150 CURSOR0,22:PRINT"                        ":
CURSOR 0,Y:PRINTS$
160 IF S$="0" THEN 220
170 CURSOR0,22:INPUT"ｷﾝｶﾞｸ ¥";T
180 CURSOR0,22:PRINT"                        ":
CURSOR20,Y:PRINT T
190 K=K-T:CURSOR30,Y:PRINT K
200 Y=Y+1:IF Y=21 THEN CLS:Y=6
210 GOTO 140
220 IFK<=0THEN PRINT"ｱｶｼﾞﾃﾞｽ !!":GOTO
240
230 PRINT "ｼﾞｮｳｽﾞﾆ ﾂｶｲﾏｼﾀ ﾈ!!!"
240 END
```

```
          コズ゛カイ チョウ
-------------------------------------------
      コンゲ゛ツ ブ゛ン ¥3000
-------------------------------------------
   コウモク              キンガ゛ク        ノコリ

マガ゛ジ゛ン                  180        2820
タイヤキ                     120        2700
デ゛ンシャチン                 340        2360
エイガ゛                     800        1560
ホット ド゛ッグ゛                250        1310
ジ゛ュース                    150        1160
プ゜ラモデ゛ル                 850         310

-------------------------------------------
シナモノ
```